LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

DTIC ACCESSION NUMBER

LEVEL

INVENTORY

DOCUMENT IDENTIFICATION

H A N D L E W I T H C A R E

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I ☒
DTIC TRAC ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DATE ACCESSIONED

Reproduced From
Best Available Copy

DATE RETURNED

19981223 033

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM 70A JUN 90

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

.14-79

TACTICAL INFORMATION EXCHANGE SYSTEM (TIES)

FINAL REPORT

CDRL ITEM #A003

Task Order No. 33

Contract N62269-78-C-0191

NADC Tech. Info.

Prepared for

NAVAL AIR DEVELOPMENT CENTER

Warminster, Pennsylvania

September 1979

NO DISTRIBUTION STATEMENT

COMPUTER SCIENCES CORPORATION

COMPUTER SCIENCES CORPORATION

DEFENSE SYSTEMS DIVISION (215) 657-4500

101 MASONS MILL BUSINESS PARK · HUNTINGDON VALLEY, PA. 19006

CBOW-1660
11 September 1979

Department of the Navy
Naval Air Development Center
Warminster, Pa. 18974

Attention: NADC/4041

Subject: Contract N62269-78-C-0191; Task Order 0033

Enclosure: CDRL Item A003; Final Technical Status Report

Gentlemen:

In accordance with the requirements of the subject contract, Computer Sciences Corporation is forwarding nine(9) copies of the enclosed report, plus one (1) copy of a tape cartridge.

This submission covers period ending August 1979. By copy of this letter the following addressees are also receiving copies of this submission.

Sincerely,

George B. Edwards

George B. Edwards
Senior Contract Administrator

GBE/jk

cc: NADC/813 w/3 encl. (Report Only)
NADC/8453 (Letter Only)
V.A. Mutinani, ACO/DCASR, DCRP/GACV

TACTICAL INFORMATION EXCHANGE SYSTEM (TIES)

FINAL REPORT

CDRL ITEM #A003

Task Order No. 33

Contract N62269-78-C-0191

Prepared for

NAVAL AIR DEVELOPMENT CENTER

Warminster, Pennsylvania

September 1979

COMPUTER SCIENCES CORPORATION

101 Masons Mill Business Park

Huntingdon Valley, Pennsylvania 19006

Major Offices and Facilities Throughout the World

TABLE OF CONTENTS

	Page
INTRODUCTION	1
APPENDIX A - PROGRAMMING GUIDELINES ANALYSIS	A-1
APPENDIX B - PRELIMINARY SOFTWARE PPS	B-1
APPENDIX C - THE INITIALIZATION CODE FOR THE SYSTEM CONTROLLER	C-1
APPENDIX D - CODE LISTING AND FLOWCHARTS	D-1

INTRODUCTION

This task order was to provide a variety of support to the TIES project.

TIES software requirements were to be identified through an analysis of the operational features of the system.

Software documentation guidelines had to be devised to track both design and software development in the context of standard military practice.

- TIES special needs are considered,
- high visability is provided to TIES management on project progress,
- the documentation plan is cost effective, and
- the documentation tends to reduce project risk.

Guidelines for the use of structured programming techniques were to be prepared.

ACCOMPLISHMENTS

For Mr. C. Nowicki, Cognizant Engineer during this task order, CSC has:

- Developed a programming guidelines analysis which discusses structured programming guidelines, identifies various proven programming languages and techniques. Further the report included an evaluation of several computer languages in terms of these criteria.

- Developed a preliminary software PPS for TIES. This report developed a preliminary functional specification, developed a detailed specification for the initialization segment of the system controller defined a Quality Assurance Plan and examined FASP as a useful QA tool.
- Coded, flowcharted, and reported on the initialization segment of the system controller.
- Developed a set of TIES software documentation guidelines.
- Analyzed the requirements for developing software in a secret environment.
- Introduced TIES personnel to the FOURTH language.

Much of this documentation is included in subsequent appendices.

APPENDIX A

PROGRAMMING GUIDELINES ANALYSIS

PROGRAMMING GUIDELINES ANALYSIS

INTRODUCTION

This paper discusses the programming guidelines that should be followed by TIES to encourage structured programming (SP) development. It also identifies proven programming methods and techniques that are language independent.

STRUCTURED PROGRAMMING - GOALS

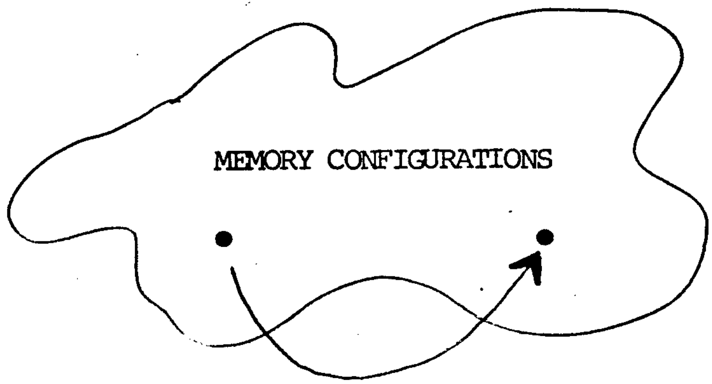
Let us begin by trying to determine what SP is trying to accomplish. Consider the following definition of programming.

Programming. The unambiguous specification of a computation to a given computer.

This is saying that programming is a process where a programmer tells a particular computer how to perform a computation. Notice the emphasis to a specific computer. The specificity is to the capabilities of the computer - two machines are equivalent (in a sense) to the programmer if they have the same memory capacities and an operationally equivalent instruction set.

Basically, a computer is a function machine whose functional domain and range is its own memory: i.e. it maps memory onto other memory configurations.

Pictorially:



Computer Operation(s)

The nature of a computer implies that a programmer must do three things to perform his task.

1. The programmer must specify the parameters of the computation in terms of memory structures that the computer can manipulate (usually numbers).
2. After step 1, the programmer has to specify the computational activity to be performed in terms of the memory structures that were chosen in the first step.
3. Finally the programmer must be able to translate the output memory structures into the computational result.

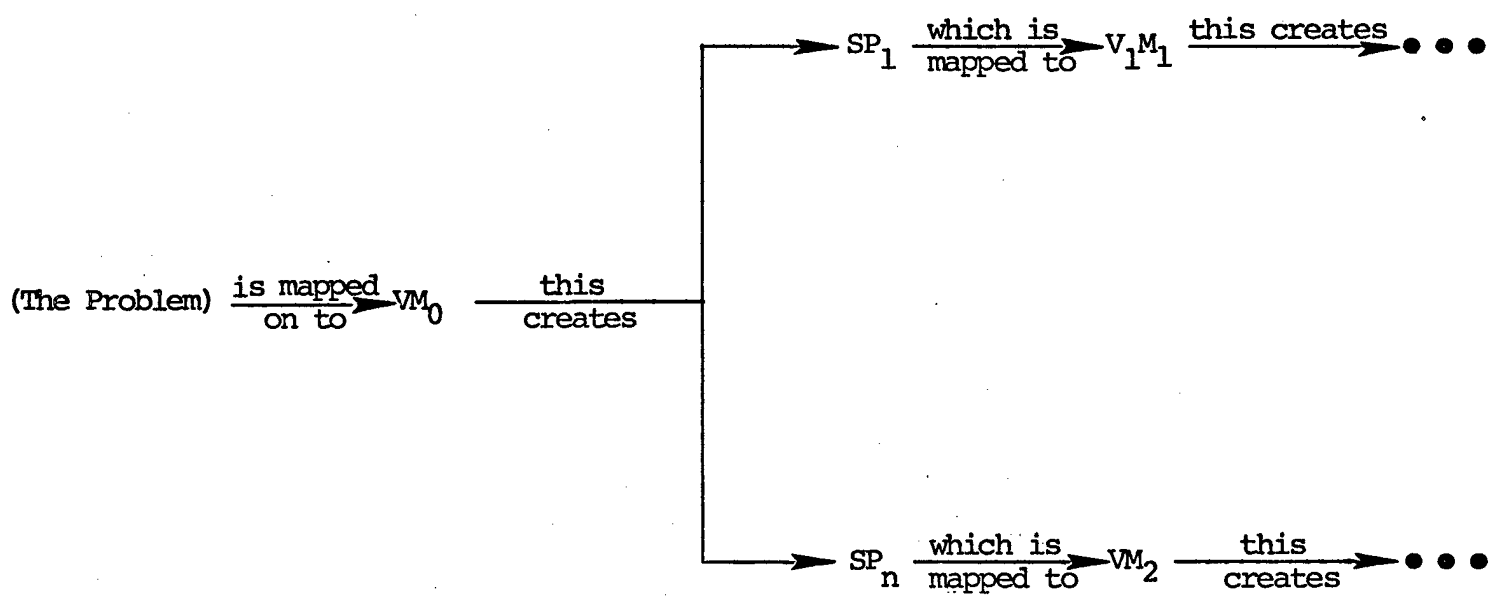
There are problems involved in the implementation of each of the three steps mentioned above. First, the translation from problem to computer representation is not straightforward. Further, there are an infinity of valid representations with no criteria as to what representation is best - or even whether a candidate representation will work. Second, given

the representation, there is no straightforward way to go from the conceptualized computation to a computation on the represented data. Finally, the first two steps don't necessarily define how the third is to be performed.

SP suggests a way to approach the task which attempts to reduce the problems. First, SP suggests that the programmer consider the aspects of the parameters to the computation that will be manipulated during the computation. Then, the programmer assumes that a computer exists that has a memory capable of representing the parameters and an instruction set that can manipulate the aspects of the parameters in the necessary way. A program is written for this "virtual" computer. (Note that it should be easier to ensure that this program performs correctly.) At this point, a program exists for the virtual computer that may or may not run on a real computer. Suppose it doesn't work on a real computer. This will occur if (1) the program is wrong, (2) some discrete portion in the program cannot be represented on the real computer, or (3) some manipulation performed in the program can't be performed on the real computer. The structured programmer now looks upon either (2) or (3) as a requirement for an additional program to be developed using SP principles. This subproblem should (hopefully) be independent of and smaller than the first original problem. Pictorially, we have:

SP DEVELOPMENT

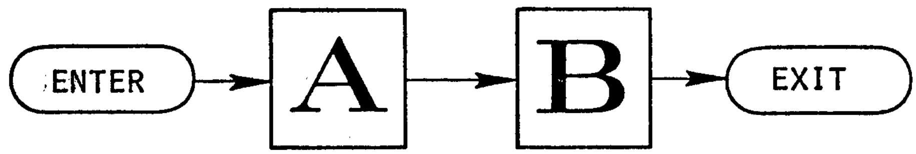
VMz = Virtual Machine
SPz = Subproblem



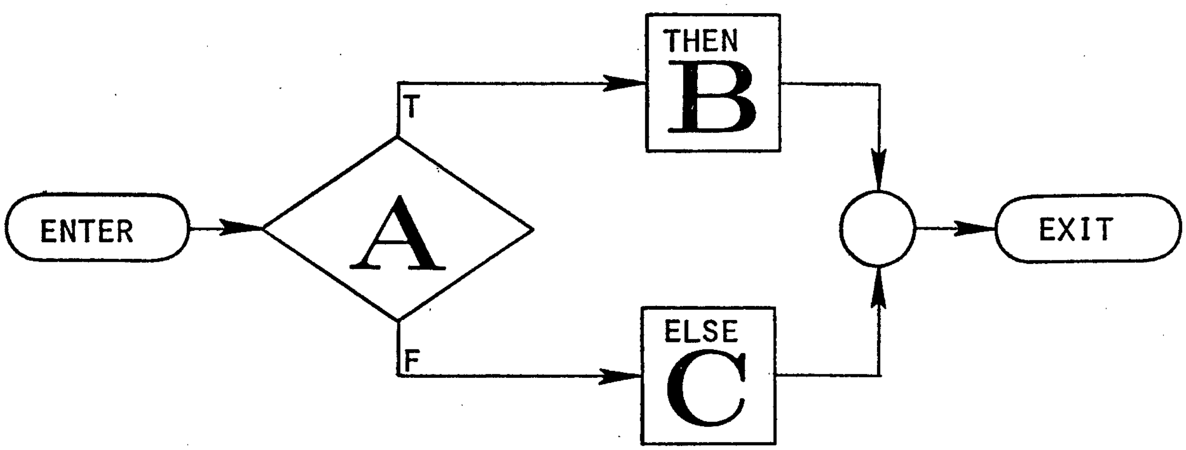
Now, the role of languages and software tools in SP becomes apparent. They are an aid (or hinderence) to SP in as much as they facilitate (or stand in the way of) mapping problems onto the virtual machines. Their abstractive power is the issue.

Note that there are two things to be abstracted, parameters (or data) and computational steps. The computational issues are well understood, and it can be shown that only a small number of ways of combining these steps (as control structures) are useful and significant. The following set of control structures are defined in a proposed MIL-STD (Military Standard for Tactical Software Development).

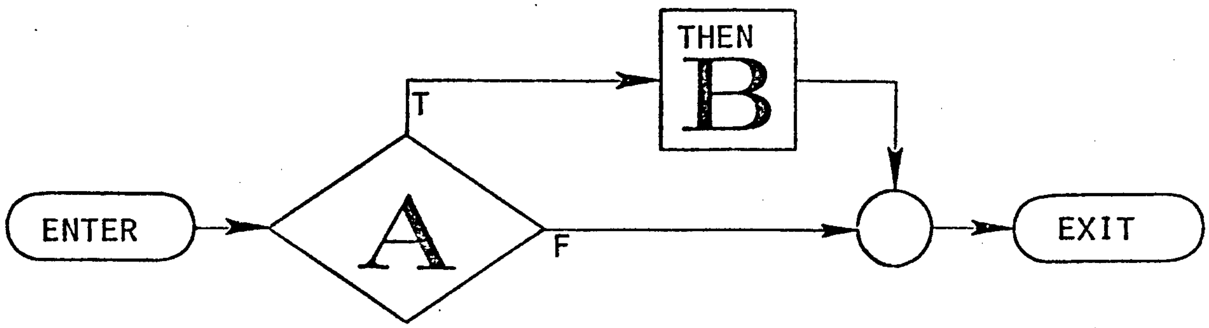
SEQUENCE



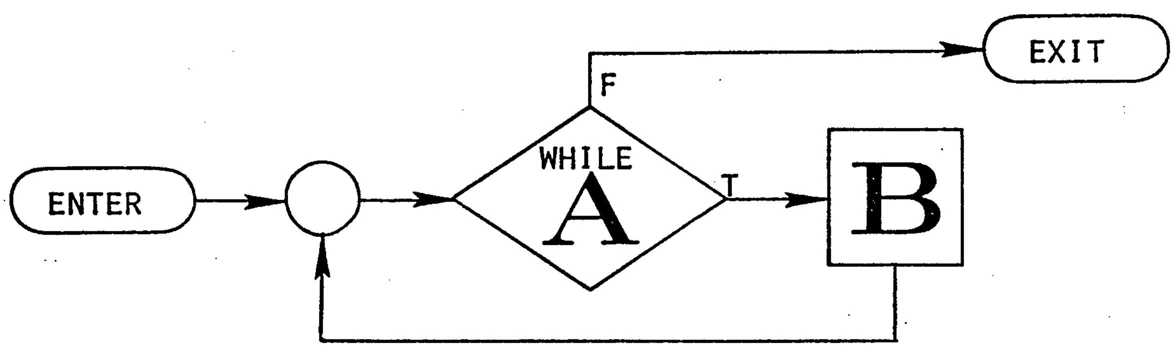
IF THEN ELSE



The flow of control will return to a common point after executing either process B or C. A predicates the conditional execution. If control is to skip a process pending the condition of A, then the flowchart can be modified thusly:

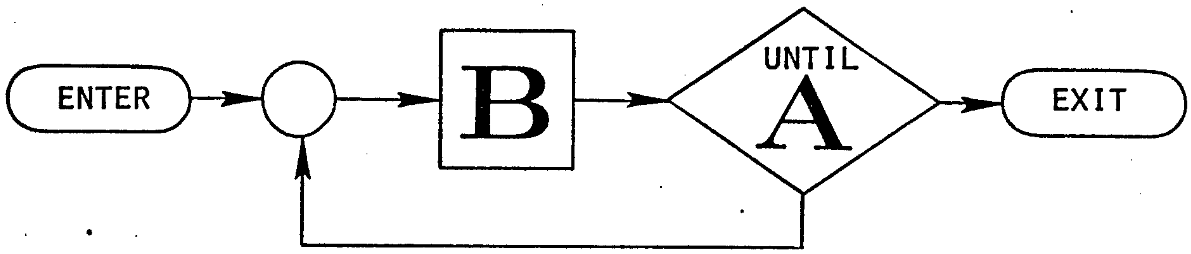


DO WHILE



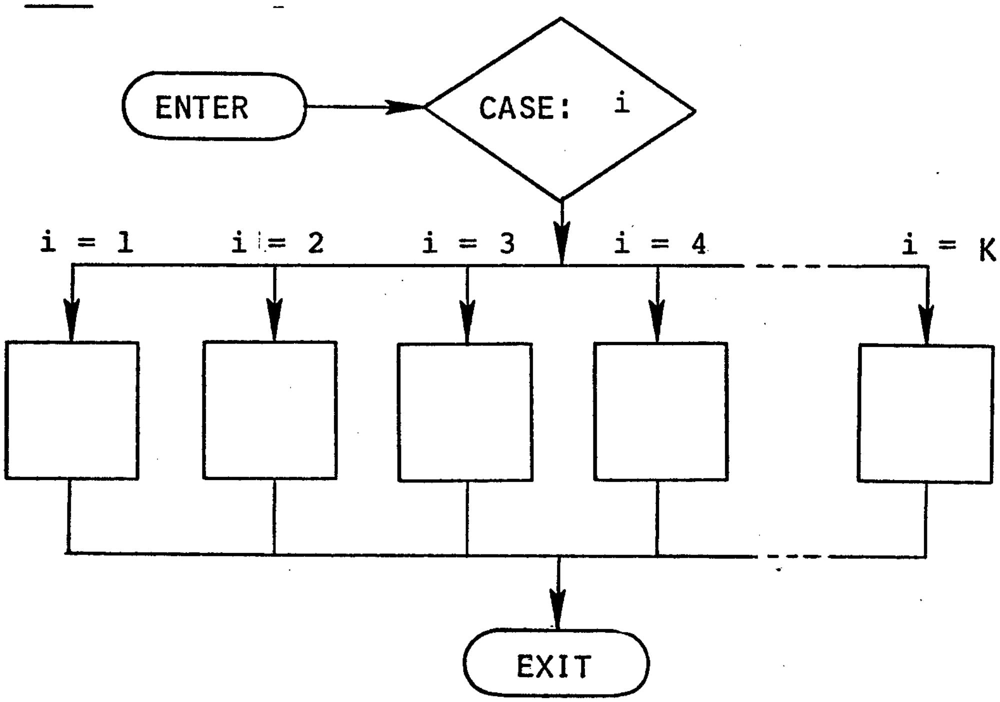
The DO WHILE structure is a loop, in which the condition A is evaluated. If found to be true, then control is passed to process B, and then condition A is evaluated again. If condition A is false then control is passed out of the loop.

DO UNTIL



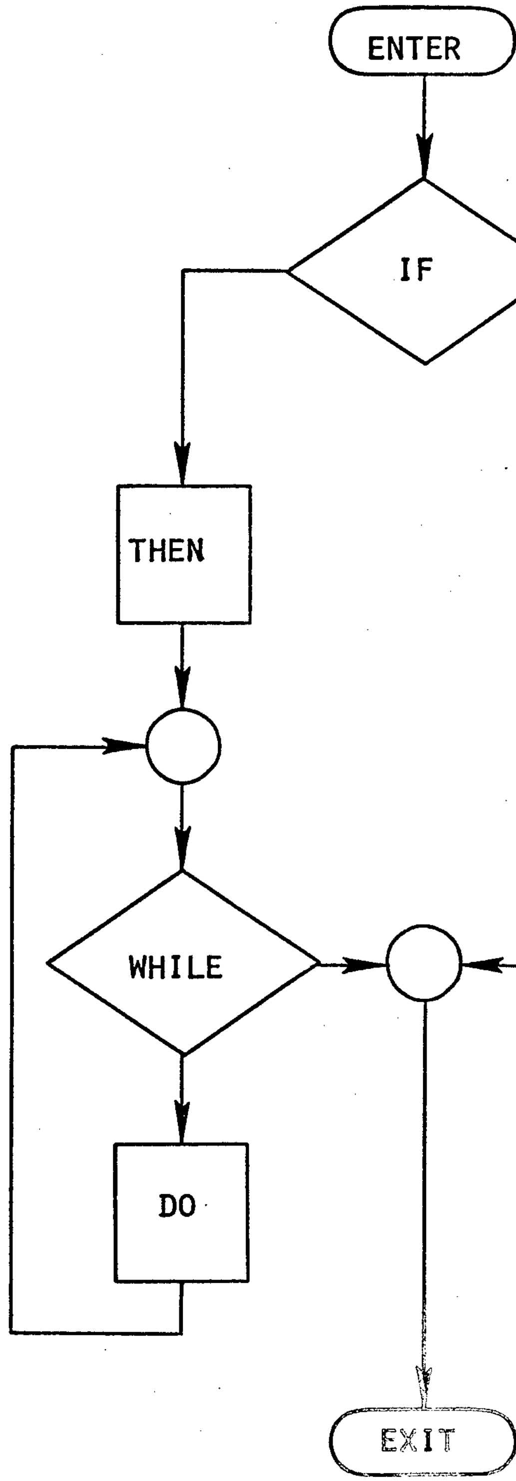
The DO UNTIL structure is similar to the DO WHILE -- except that the test of condition A is performed after process B has executed. Thus the DO UNTIL loop will be performed once regardless of the value of condition A.

CASE



Control is passed to process 'K' based on the value of i. Note that this definition is recursive in nature; structured programs of any degree of complexity can be built up, if they can be broken down into individual components. The CASE structure is a generalization of the IF THEN and the IF THEN ELSE.

The following is an example of such a recursive structure. If a condition holds, a DO WHILE loop is executed, otherwise the loop is not performed.



These structures should be the only ones used regardless of the language chosen.

IRONMAN specification for computational control structures added some relevant requirements (listed below).

IRONMAN CONTROL STRUCTURE REQUIREMENTS

Sequential Control. There shall be a sequential control mechanism (i.e., a mechanism for sequencing statements). Explicit statement delimiters shall be required. (Note the choice between terminators and separators can be left to the user.) This amplifies the SEQUENCE structure previously described.

Conditional Control. There shall be conditional control structures that permit selection among alternative control paths. The selected path may depend on the value of a conditional expression, on a computed choice among labeled alternatives, or on the true condition in a set of mutually exclusive conditions. The control action must be specified for all values of the discriminating condition.

Short Circuit Evaluation. There shall be forms for short circuit conjunction and disjunction of Boolean expressions in conditional and iterative control structures. A short circuit evaluation suspends evaluation of a Boolean expression as soon as the value of the expression is known.

Iterative Control. There shall be an iterative control structure that permits a loop to have several explicit termination conditions and

permits termination anywhere in the loop. Iterative control structures may be entered only at the head of the loop.

Consider the following example:

A FOR loop is an example of an iterative control structure in BASIC. The FOR statement through the associated NEXT statement constitute the body or scope of the loop. Such a loop has an explicit termination condition - the loop is exited after it has been performed the proper number of times. Since a programmer can leave the loop by a GO TO statement - termination is permitted anywhere in the loop. However, there is no restriction that the scope of the loop can be entered only at the FOR statement.

Loop Control Variables. Loop control variables, if any, shall be local to the iterative control statement. Assignment shall not be allowed to control variables from the loop body. It shall be possible to iterate over sequences of integers and over elements of an enumeration type.

Explicit Control Transfer. There shall be an explicit mechanism for control transfer (i.e., the GO TO). However, the mechanism shall not permit:

- Transfers out of declarations, functions, procedures, encapsulated definitions, (see the discussion of encapsulation in the next section).

- Transfers into narrower scopes,
- Transfers into control structures,
- Transfers in the form of switches, (the computed GO TO of FORTRAN), designational expressions, label variables, label parameters, or alter statements.

The IRONMAN standards also listed relevant requirements for data abstraction language features and other, more general purpose, abstraction features.

IRONMAN DATA ABSTRACTION FEATURES

Boolean Types. There shall be a predefined unordered enumeration type. The Boolean type shall have operations for conjunction, inclusive disjunction, and negation.

Composite Type Definitions. It shall be possible to define types that are Cartesian products of other types. Composite types shall include arrays (i.e., composite data with indexible components of homogeneous types) and records (i.e., composite data with labeled components of heterogeneous type).

Component Specifications. For elements of composite types, the type of each component (i.e., field) must be explicitly specified in programs and determinable at translation time. Components may be of any type (including array and record types). Range, precision and scale specifications shall be required for each component of appropriate numeric types.

Operations on Composite Types. A value accessing operation shall be automatically defined for each component of composite data elements. Assignment shall be automatically defined for components that have alterable values. A constructor operation (i.e., an operation that constructs an element of a type from its constituent parts) shall be automatically defined for each composite type. An assignable component may be used anywhere in a program that a variable of the component's type is permitted.

Array Specifications. The number of dimensions for each array must be specified in programs and shall be determinable at translation time. The range of subscript values for each dimension must be specified in

programs and shall be determinable by the time of array allocation. The range of subscript values shall be restricted to a contiguous sequence of integers or to a contiguous sequence from an enumeration type.

Operations on Subarrays. There shall be built-in operations for value access, assignment, and concatenation of contiguous sections of one-dimensional arrays of the same component type.

Operations on Records. The assignment operation is to be defined between variables of type record, where both record definitions have identical names and components.

Encapsulated Definitions. It shall be possible to encapsulate definitions. Encapsulations may contain definitions of the data elements comprising a type and of operations.

Effect of Encapsulation. The effect of encapsulation shall be to inhibit external access to implementation properties of the definition. In particular, declarations made within an encapsulation shall not automatically be accessible outside the encapsulation. Data elements defined in an encapsulation shall not automatically inherit the operations of the types with which they are represented.

Own Variables. It shall be possible within encapsulations to declare variables that are accessible only within the encapsulation but remain allocated throughout the scope in which the encapsulation is declared. Such variables shall retain their values between entries to the encapsulation. It shall be possible to initialize such variables at the time of their apparent allocation.

Operations Between Types. It shall be possible to define operations, like type conversion, that require access to local properties of more than one encapsulated definition. (Note that this capability violates the purpose of encapsulation and thus its use should be avoided wherever possible.)

(The general purpose abstraction mechanisms follow.)

IRONMAN GENERAL PURPOSE ABSTRACTION MECHANISMS

Function and Procedure Definitions. Functions (which return values to expressions) and procedures (which can be called as statements) shall be definable in programs. Existing functions (including those called using infix forms) and procedures shall be extensible to new data types (i.e., overloading shall be permitted).

Function Declarations. The result type for each function must be explicitly specified in the function declaration and shall be determinable at translation time. A function of two arguments may be specified as

associative in this declaration. (Note that the latter requirement reduces the need for explicit parentheses.)

Restrictions on Functions. A function may only have input parameters and may not be called in a scope that contains variables that are referenced or assigned directly or indirectly within the body of the function.

Formal Parameter Classes. There shall be three classes of formal parameters: 1) input parameters, which act as constants that are initialized to the value of corresponding actual parameters at the time of call, 2) input-output parameters, which enable access and assignment to the corresponding actual parameters, and 3) output parameters, which act as local variables whose values are transferred to the corresponding actual parameter only at the time of normal exit.

Parameter Specifications. The type of each formal parameter must be explicitly specified in programs and shall be determinable at translation time. Parameters may be of any type. Range, precision, and scale specifications shall be required for each formal parameter of appropriate numeric types. A translation time error shall be reported wherever corresponding formal and actual parameters are of different types and wherever a program attempts to use a constant or an expression where a variable is required.

Formal Array Parameters. The number of dimensions for formal array parameters must be specified in programs and shall be determinable at translation time. Determination of the subscript range for formal array parameters may be delayed until execution and may vary from call to call. Subscript ranges shall be accessible within function and procedure bodies without being passed as an explicit argument.

Restrictions to Prevent Aliasing. Aliasing (i.e., multiple access paths to the same variable from a given scope) shall not be permitted. In particular, a variable may not be used as two output arguments in the same call to a procedure, and a nonlocal variable that is accessed or assigned with a procedure body may not be used as an output argument to that procedure.

ADDITIONAL (NON-SP) CRITERIA

There are additional (non-SP) considerations that must be examined in any language choice for TIES. They are listed below.

TIES NON-SP LANGUAGE CONSIDERATIONS

Availability. Working compilers must exist which execute on a machine available to TIES software personnel. They must exist at least one year before projected software startup.

Portability Costs. TIES must have a clear understanding of the software costs associated with moving a program from one processor to another.

Convenience. Compilers for the language must be easy for software development and maintenance personnel to use.

Programmer Base. Programmers must exist who understand the language (or a similar language) and the kind of problems TIES is trying to solve.

Generality. The ideal TIES language would be applicable to all processors within the system.

Languages To Be Considered. Four languages are to be considered here. They are:

1. ADA - The DoD-1 Language (Described in later section)
2. BASIC - A language developed for on-line programming.
3. FORTH - A language developed for the programming of micro-processors and minicomputers.
4. PASCAL - The first language designed to support SP.

The following tables evaluate each language according to the criteria previously described. The weighting factor reflects my accessment of the relative importance of each feature. Raw scores are numerical

value from 0 to 1 and indicate the degree of satisfaction of the feature. (1 means full satisfaction.) Table 6 integrates the 5 previous tables.

TABLE 1 - MEASUREMENT OF THE LANGUAGES IN TERMS OF CONTROL STATEMENTS

*Estimated

FEATURE	WEIGHTING FACTOR	NORMALIZED WEIGHTING FACTOR	RAW SCORES*				NORMALIZED WEIGHTED SCORES			
			ADA	BASIC	FORTH	PASCAL	ADA	BASIC	FORTH	PASCAL
IF ... THEN (...ELSE)	1	0.08	1.00	0.25	0.75	1.00	0.08	0.02	0.06	0.08
DO ... WHILE	4	0.31	1.00	0.00	0.75	1.00	0.31	0.00	0.23	0.31
DO ... UNTIL	4	0.31	0.00	0.00	0.75	0.00	0.00	0.00	0.23	0.00
CASE	4	0.31	1.00	0.25	0.10	1.00	0.31	0.08	0.03	0.31
	13	1.00	3.00	0.50	2.35	3.00	0.70	0.10	0.55	0.70

TABLE 2 - MEASUREMENT OF THE LANGUAGES IN TERMS OF CONTROL STRUCTURES

*Estimated

FEATURE	WEIGHTING FACTOR	NORMALIZED WEIGHTING FACTOR	RAW SCORES*				NORMALIZED WEIGHTED SCORES			
			ADA	BASIC	FORTH	PASCAL	ADA	BASIC	FORTH	PASCAL
Sequential Control	4	0.22	1.00	1.00	1.00	1.00	0.22	0.22	0.22	0.22
Conditional Control	4	0.22	1.00	0.75	0.50	1.00	0.22	0.17	0.11	0.22
Short Circuit Evaluation	1	0.06	1.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00
Iterative Control	4	0.22	1.00	0.00	0.00	0.00	0.22	0.00	0.00	0.00
Loop Control Variables	2	0.11	1.00	0.00	0.00	1.00	0.11	0.00	0.00	0.11
Explicit Control Transfer	3	0.17	1.00	0.00	0.00	0.75	0.17	0.00	0.00	0.13
	18	1.00	7.00	1.75	1.50	3.75	1.00	0.39	0.33	0.68

TABLE 3 - MEASUREMENT OF THE LANGUAGES IN TERMS OF DATA ABSTRACTION FEATURES

*Estimated

FEATURE	WEIGHTING FACTOR	NORMALIZED WEIGHTING FACTOR	RAW SCORES* ADA	BASIC	FORTH	PASCAL	NORMALIZED WEIGHTED SCORES ADA	BASIC	FORTH	PASCAL
Boolean Types	5	0.17	1.00	0.00	1.00	1.00	0.17	0.00	0.17	0.17
Composite Type Definitions	3	0.10	1.00	0.00	0.00	1.00	0.10	0.00	0.00	0.10
Component Specifications	2	0.07	1.00	0.00	0.00	0.90	0.07	0.00	0.00	0.06
Operations on Composite Types	2	0.07	1.00	0.00	0.00	0.90	0.07	0.00	0.00	0.06
Array Specifications	6	0.21	1.00	0.80	0.00	0.90	0.21	0.17	0.00	0.19
Operations on Subarrays	4	0.14	1.00	0.70	0.00	0.00	0.14	0.10	0.00	0.00
Operations on Records	3	0.10	1.00	0.00	0.00	1.00	0.10	0.00	0.00	0.10
Encapsulated Definitions	1	0.03	1.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
Effect of Encapsulation	1	0.03	1.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
Own Variables	1	0.03	1.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
Operations Between Types	1	0.03	1.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
	29	1.00	11.00	1.50	1.00	5.70	1.00	0.27	0.17	0.68

TABLE 4 - MEASUREMENT OF THE LANGUAGES IN TERMS OF GENERAL PURPOSE ABSTRACTION MECHANISMS

*Estimated

FEATURE	WEIGHTING FACTOR	NORMALIZED WEIGHTING FACTOR	RAW SCORES*				NORMALIZED WEIGHTED SCORES			
			ADA	BASIC	FORTH	PASCAL	ADA	BASIC	FORTH	PASCAL
Function and Procedure Definitions	14	0.27	1.00	0.25	0.85	0.75	0.27	0.07	0.23	0.20
Function Declarations	13	0.25	1.00	0.00	0.85	0.75	0.25	0.00	0.21	0.19
Restrictions on Functions	3	0.06	1.00	1.00	0.00	1.00	0.06	0.06	0.00	0.06
Formal Parameter Classes	2	0.04	1.00	0.25	0.75	0.75	0.04	0.01	0.03	0.03
Parameter Specifications	11	0.21	1.00	0.00	0.00	0.85	0.21	0.00	0.00	0.18
Formal Array Parameters	7	0.13	1.00	0.00	0.00	0.75	0.13	0.00	0.00	0.10
Restrictions to Prevent Alligning	2	0.04	1.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
	52	1.00	7.00	1.50	2.45	4.85	1.00	0.13	0.47	0.76

TABLE 5 - MEASUREMENT OF THE LANGUAGES IN TERMS OF NON-SP CONSIDERATIONS

*Estimated

FEATURE	WEIGHTING FACTOR	NORMALIZED WEIGHTING FACTOR	RAW SCORES*				NORMALIZED WEIGHTED SCORES			
			ADA	BASIC	FORTH	PASCAL	ADA	BASIC	FORTH	PASCAL
Availability	3	0.16	0.10	1.00	1.00	0.66	0.02	0.16	0.16	0.10
Portability Costs	5	0.26	0.25	0.75	1.00	0.50	0.07	0.20	0.26	0.13
Convenience	5	0.26	0.50	1.00	0.75	0.50	0.13	0.26	0.20	0.13
Programmer Base	5	0.26	0.25	1.00	0.50	0.50	0.07	0.26	0.13	0.13
Generality	1	0.05	1.00	0.50	0.75	1.00	0.05	0.03	0.04	0.05
	19	1.00	2.10	4.25	4.00	3.16	0.33	0.91	0.79	0.55

TABLE 6 - MEASUREMENT OF THE LANGUAGES IN TERMS OF OVERALL FEATURES

*Estimated

FEATURE	WEIGHTING FACTOR	NORMALIZED WEIGHTING FACTOR	RAW SCORES*				NORMALIZED WEIGHTED SCORES			
			ADA	BASIC	FORTH	PASCAL	ADA	BASIC	FORTH	PASCAL
Control Statements	2	0.12	0.70	0.10	0.57	0.70	0.08	0.01	0.07	0.08
Control Structures	2	0.12	1.00	0.39	0.39	0.68	0.12	0.05	0.05	0.08
Data Abstraction Features	3	0.18	1.00	0.27	0.17	0.68	0.18	0.05	0.03	0.12
General Purpose Abstraction	3	0.18	1.00	0.13	0.47	0.76	0.18	0.02	0.08	0.13
Non SP Considerations	7	0.41	0.33	0.91	0.79	0.55	0.14	0.37	0.33	0.23
	17	1.00	4.03	1.80	2.39	3.37	0.69	0.50	0.55	0.64

A statistical analysis of the four normalized weighted scores suggests we cannot reject the hypothesis that all the languages are equivalent with 90% certanty. Weighing "Non SP Considerations" with integer values less than 7 imply acceptance of ADA as superior with 90% confidence (but rejection at the 95% level).

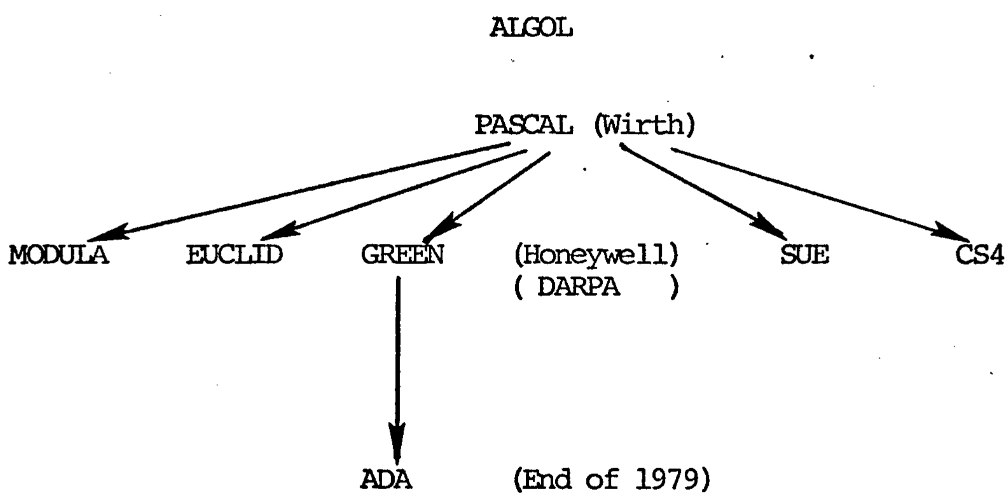
TABLE 7 - ADJUSTING NON-SP WEIGHING FACTOR

WEIGHING FACTOR	μ	σ	$\mu + \frac{1.638\sigma}{2}$	$\mu + \frac{2.353\sigma}{2}$	ADA'S SCORE
0	.57	.33	.81	.92	.94
1	.57	.27	.79	.88	.88
2	.58	.23	.76	.84	.84
3	.58	.19	.74	.80	.80
4	.59	.16	.72	.77	.77
5	.59	.13	.70	.75	.74
6	.60	.10	.68	.74	.71

ADA - THE NEW "DOD-1" LANGUAGE

This section discusses the new 'DoD-1' language and considers the efficiency of the language for TIES use. Background and current information on the language is from Lt. Col. W. Whitacker of DARPA who acts as a Special Assistant to the Director. The new language now has a name - ADA. It will be developed by Honeywell Bull out of their Green programming language. It's genealogy is indicated in the figure below. The language (not a compiler for the language) is currently in a Test and Evaluation stage and will remain so throughout 1979. Final change proposals for the language will be submitted to Honeywell in early December.

GENEALOGY OF ADA



Whitacker says that a fair amount of programming is already being done in the language to get some experience in the usefulness of the design. He knows of twenty ongoing compiler efforts in the language and suspects there are more. He knows of funded efforts for compilers that will generate code for the PDP-11, IBM 360, PDP-10, and a AYK-15(A). He had heard of a Navy effort to develop a single compiler for SPL-1, CMS-2, and ADA. Currently, a "prototype translator" for the language exists: it is written in PL-1, runs on a MULTEX system, and is available through ARPANET, TIMESHARE, and TIMENET. There are no plans to transport this program to other machines.

Whitacker knows of no funded efforts to create a compiler for ADA that will generate code for a microprocessor. However, he indicated that a Capt. Bloden at Egland was interested in such a project. Whitacker expects that such a compiler could be developed. Further, he feels that a properly designed version of this compiler could be retargeted (made to generate code for another microprocessor) at a cost of about ½ a man year. He indicated that he expects such a compiler to be available within 2-4 years.

ADA will be added to the 5000.31 approved list of languages in the spring of 1980. He feels that other languages will drop from the approved list within less than 5 years after ADA's introduction. This would, in effect, make ADA the de-facto standard for embedded computer applications for DoD and NATO. This standardization to ADA could encourage manufacturers to make processors that were oriented to the language and would facilitate its' use.

Whitacker suggests that several projects at Warminster plan to use ADA. At the time of our conversation, he did not have the information available to discuss specifics. He remembered one project as involving the development of an advanced trainer and simulator, and another as involving a fleet level upgrade of a fighter aircraft. He thought that V/STOL would use ADA as their development language.

APPENDIX B

PRELIMINARY SOFTWARE PPS

SECTION 1. SCOPE

1.1 Purpose

This document is a preliminary version of a Program Performance Specification (PPS) for TIES. It is being developed in partial fulfillment of a TIES Software Task Order under Contract N62269-78-C-0191 and addresses requirement (a) of Section 2.0 of the Statement of Work for the task order. The document reflects the software requirements for TIES in the context of TIES design of 21 March 1979.

1.2 Mission

The mission of TIES is described in the Statement of Work which initiated this document.

> "The TIES project aims at cost effective implementation of CNI functional capabilities of a platform by means of an integrated systems approach that is based upon the use of a defined set of standardized modular resources."

For further information see paragraph 1.1.3 of TIES SOFTWARE ORIENTATION AND REQUIREMENTS. (TSO & R)

1.3 Scope

This document is a preliminary draft of the contents of a PPS to be eventually produced for TIES. It is not intended to be complete: the software component(s) of TIES are still under analysis. This document will define the design as it is understood 21 March 1979. The document can then be used as a basis for subsequent analysis and development.

1.3.1 Identification

Appendix B of the specification lists approved identification, nomenclature, and authorized abbreviations for TIES.

1.3.2 Functional Summary

The TIES architecture identifies six software areas by type. They are listed below:

TIES SOFTWARE AREAS

TYPE 1 -	System Controller (SysC)
TYPE 2 -	Satellite Controller (SatC)
TYPE 3 -	External Test (ExtT)
TYPE 4 -	Data Processor (DatP)
TYPE 5 -	Wideband Signal Conversion Unit (WBSCU)
TYPE 6 -	Narrowband Signal Conversion Unit (NBSCU)

Figure 1.3.1 is a schematic of the major system components (excluding the External Test System). The solid lines represent communications paths between components. Figure 3.3.1 is a top-level block diagram of the system.

The Software Area comprising the SysC is identified by the topmost box in the diagram and circled in Figure 1.3.2. It is the standard system control interface point and can access (indirectly) all system components through SatCs. Requirements for the area are discussed in more detail in Table 1.3.1.

The Software Area comprising the SatC is identified by the wideband and narrowband Satellite Controller boxes and by the Narrowband site boxes. They are circled in Figure 1.3.3. SatCs are the control interface between the SysC and the Frequency and Signal Conversion and distribution subsystems within TIES. Functionally, they exist to (1) provide control backup in the event of SysC failure, (2) control conversion subsystem initialization and tuning, (3) performance monitoring, and (4) perform detailed sequencing required for Built-In Test (BIT) activities. Requirements for the area are discussed in more detail in Table 1.3.2.

The ExtT System for TIES interfaces with the subsystem FDM through bus couplers to WBSCUs, NBSCUs, and both Wideband and Narrowband Sites. The interface is diagrammed in Figure 1.3.4. The FDM subsystem is referred to as the Signal Distribution Subsystem of TIES. This Subsystem, plus the Frequency Conversion Subsystem and the Signal Conversion Subsystem are circled in the figure. Additional interfaces for the ExtT System exist within TIES. The ExtT System can replace the SysC. Further, the ExtT system can be coupled to an antenna and it's associated frequency conversion site. The ExtT System performs both on-line and off-line tests and is discussed more fully in Table 1.3.3.

The Software Area comprising the DatP is identified by the Data Processor boxes which interface with the WBSC and the NBSC respectively. Both function as interfaces between TIES and human support devices for the various waveforms handled by the system. Message formatting/decomposition and EDAC also occur here. Requirements for the area are discussed in more detail in Table 1.3.4.

The Software Area comprising the WBSCU is identified by the WBSCU itself, its associated FDM Channel Selector, a Rapid Remote Control Device (RRCD), and a Wideband Site controlled by the RRCD. They are circled in Figure 1.3.6.

The WBSC system handles transmission and/or reception of JTIDS, GPS, TACAN, and IFF/DABS waveforms. Requirements for the area are discussed in more detail in Table 1.3.5. Figure 3.3.4 provides a block diagram.

The Software Area comprising the NBSC consists of the NBSCU and its associated FDM Channel Selector. They are circled in Figure 1.3.7. The NBSC System handles transception of AM, FM, FSK, SSB, and DCPSK waveforms. Two NBSCU's will exist in a given TIES system. This provides a graceful degredation capacity in the event of failure of one of the units. Requirements for the area are discussed in more detail in Table 1.3.6. Figure 3.3.2 provides a block diagram.

Figure 1.3.1. TIES SOFTWARE SCHEMATIC (EXCLUDING EXTERNAL TEST SYSTEM)

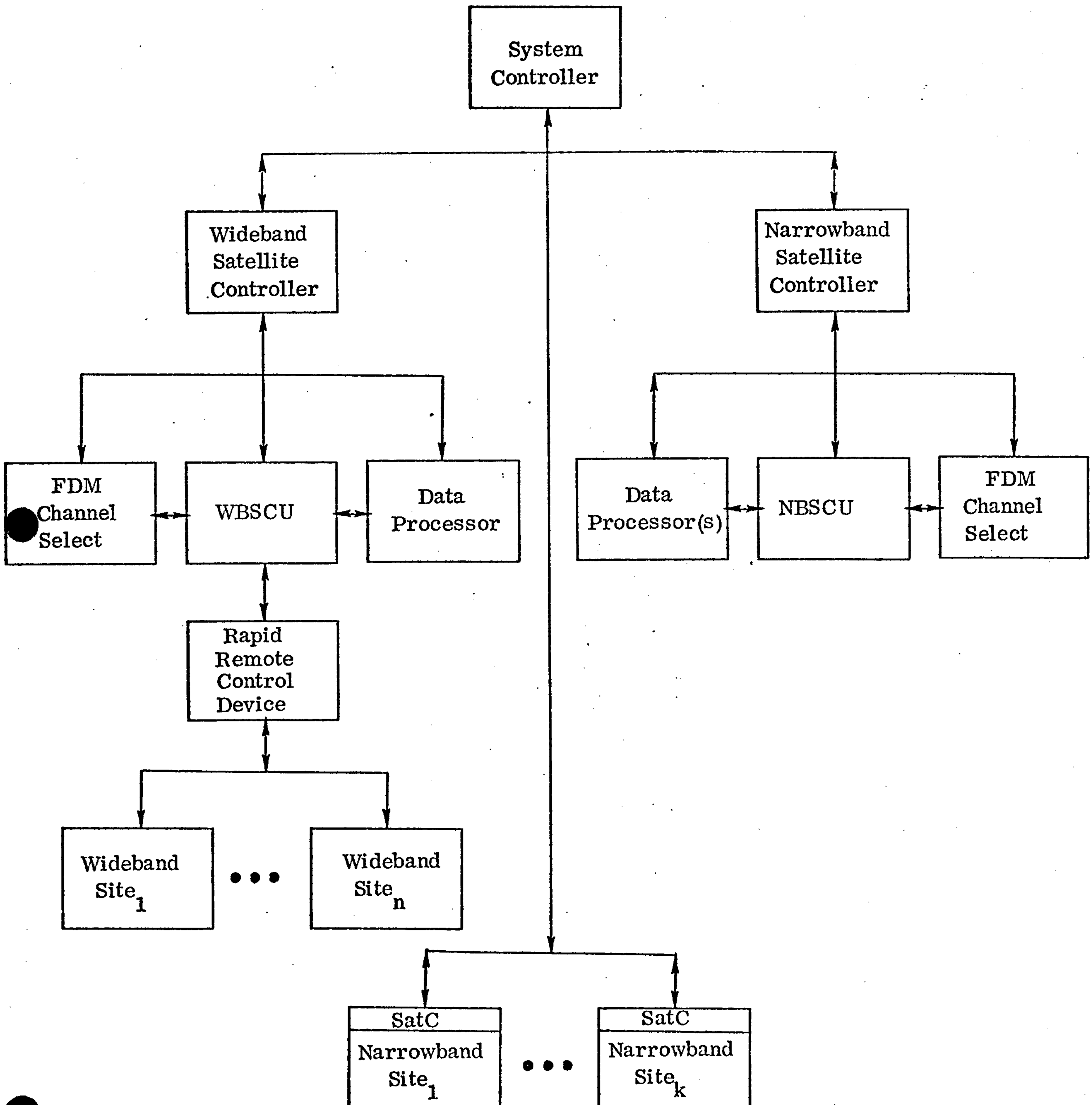


Figure 1.3.2. TIES SYSTEM CONTROLLER AREA

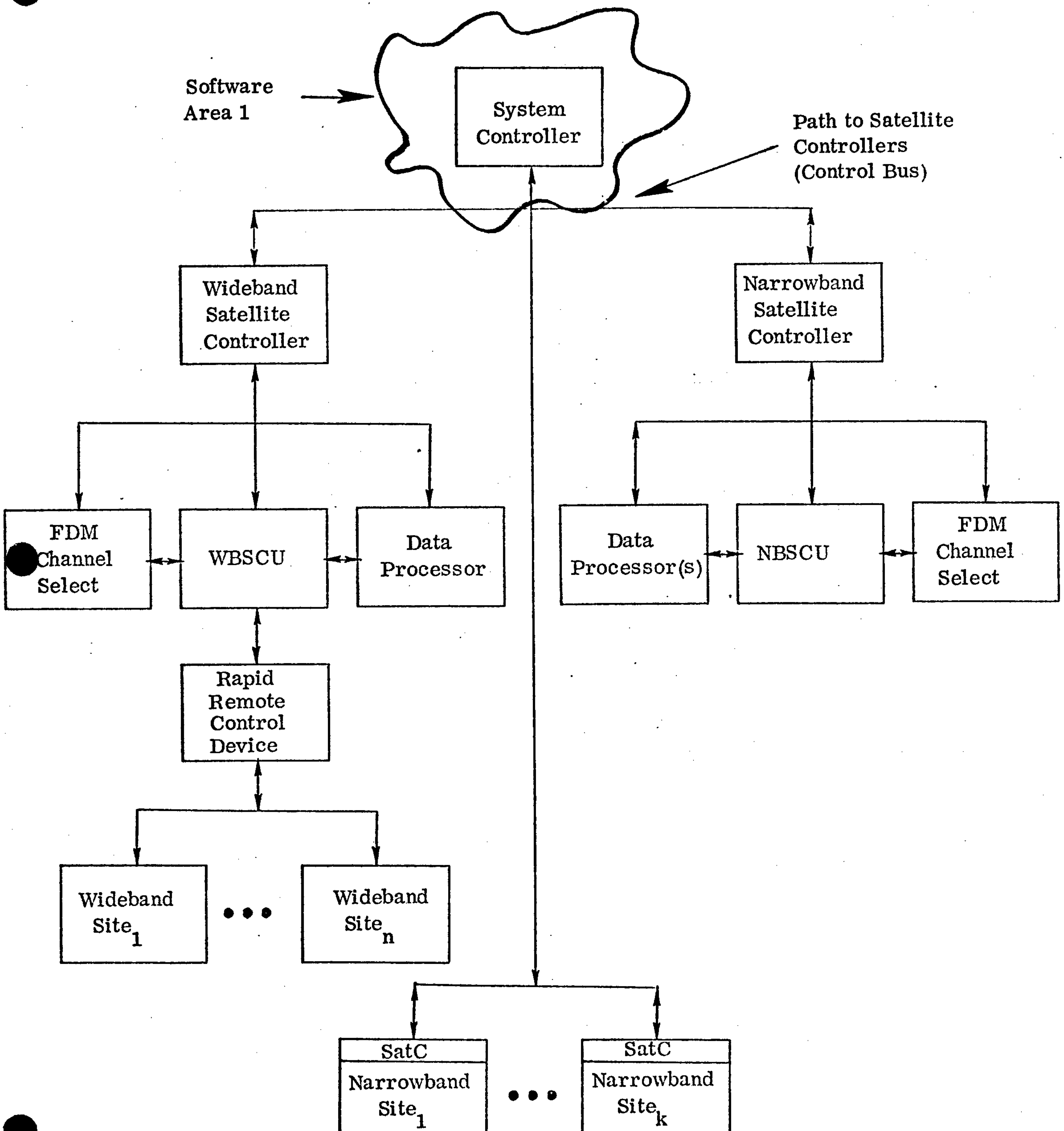


Figure 1.3.3. TIES SATELLITE CONTROLLER AREA

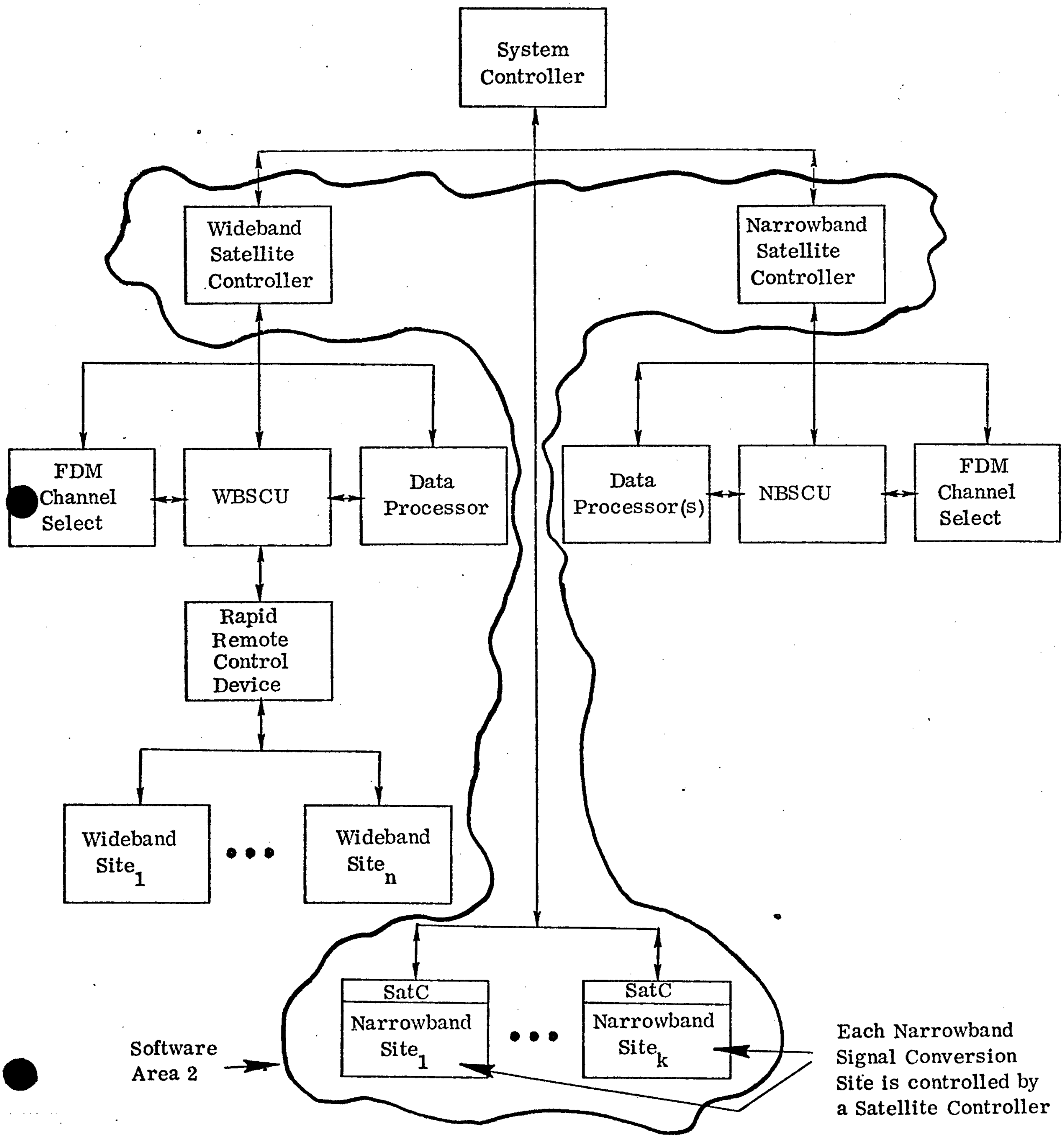


Figure 1.3.4. TIES EXTERNAL TEST SYSTEM INTERFACE

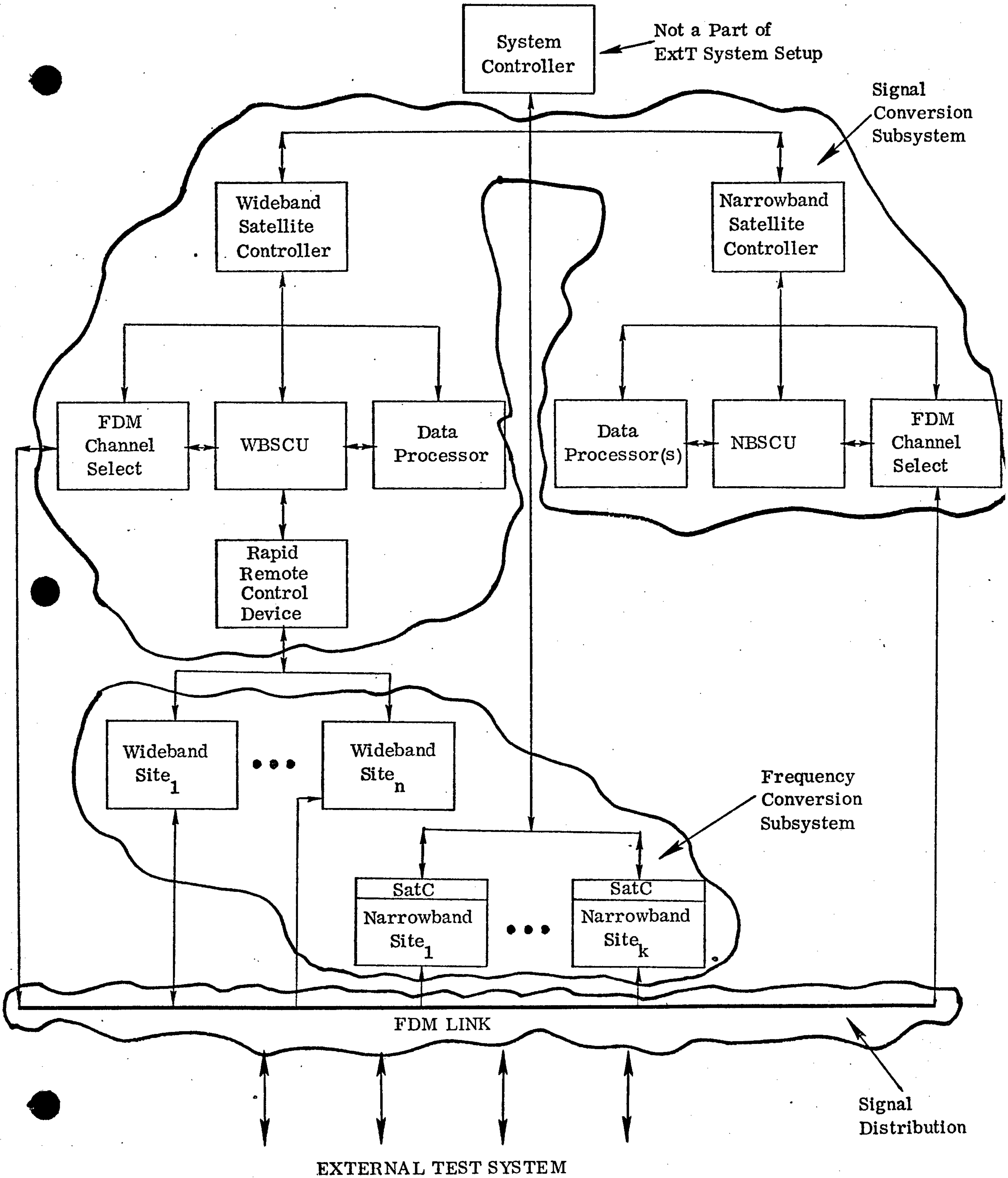


Figure 1.3.5. TIES DATA PROCESSOR AREA

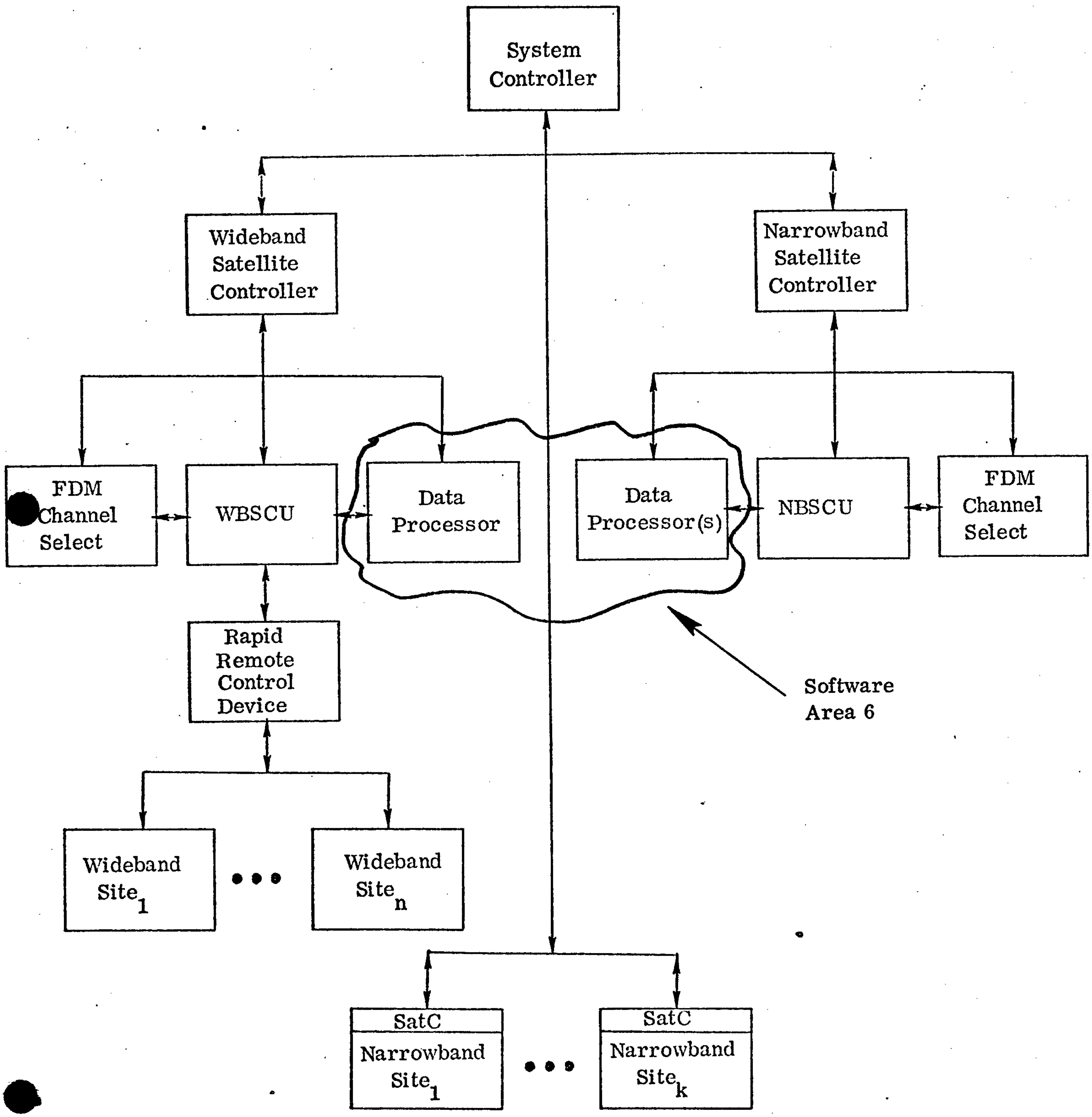


Figure 1.3.6. TIES WIDEBAND SIGNAL CONVERSION AREA

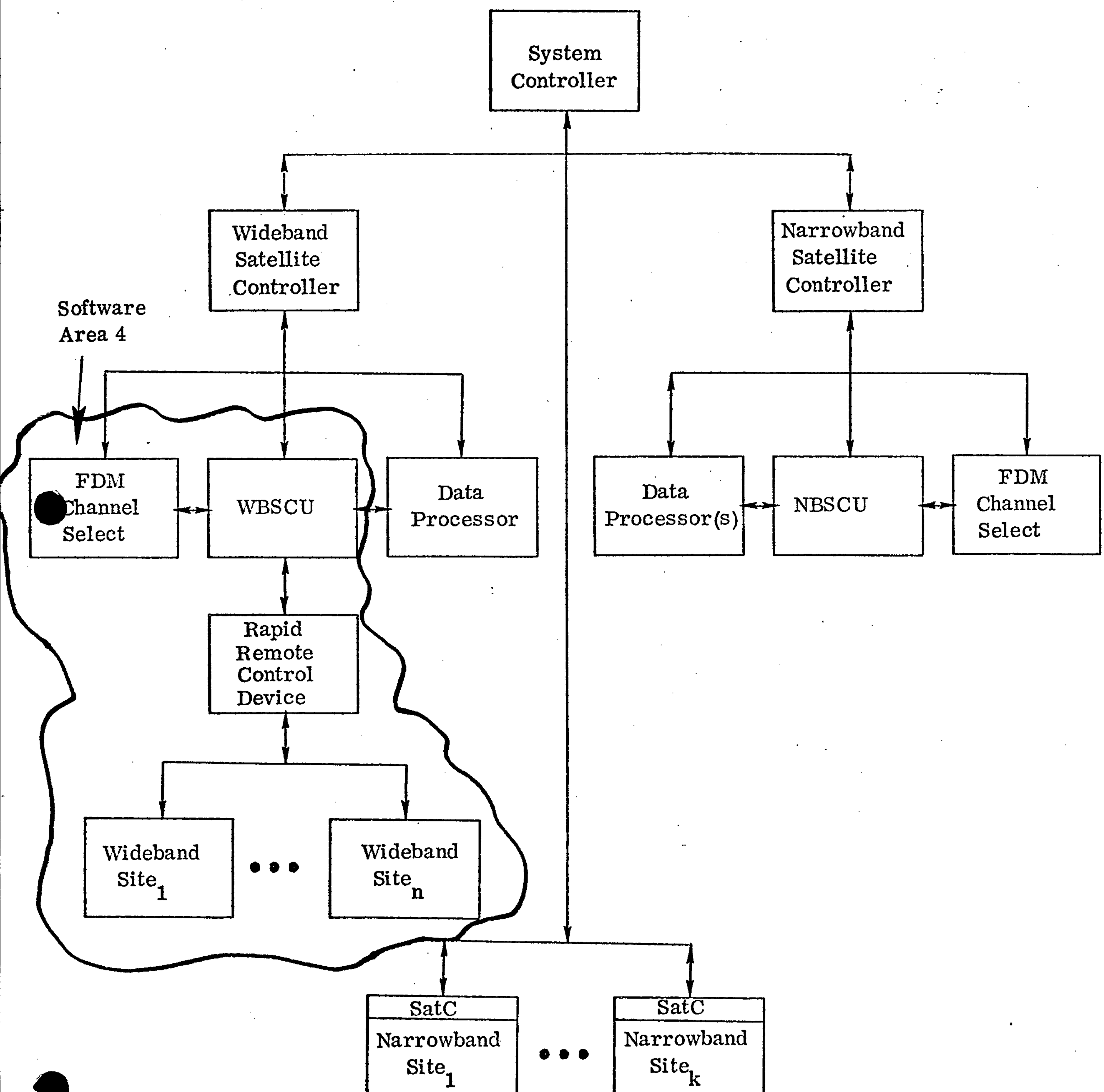


Figure 1.3.7. TIES NARROWBAND SIGNAL CONVERSION AREA

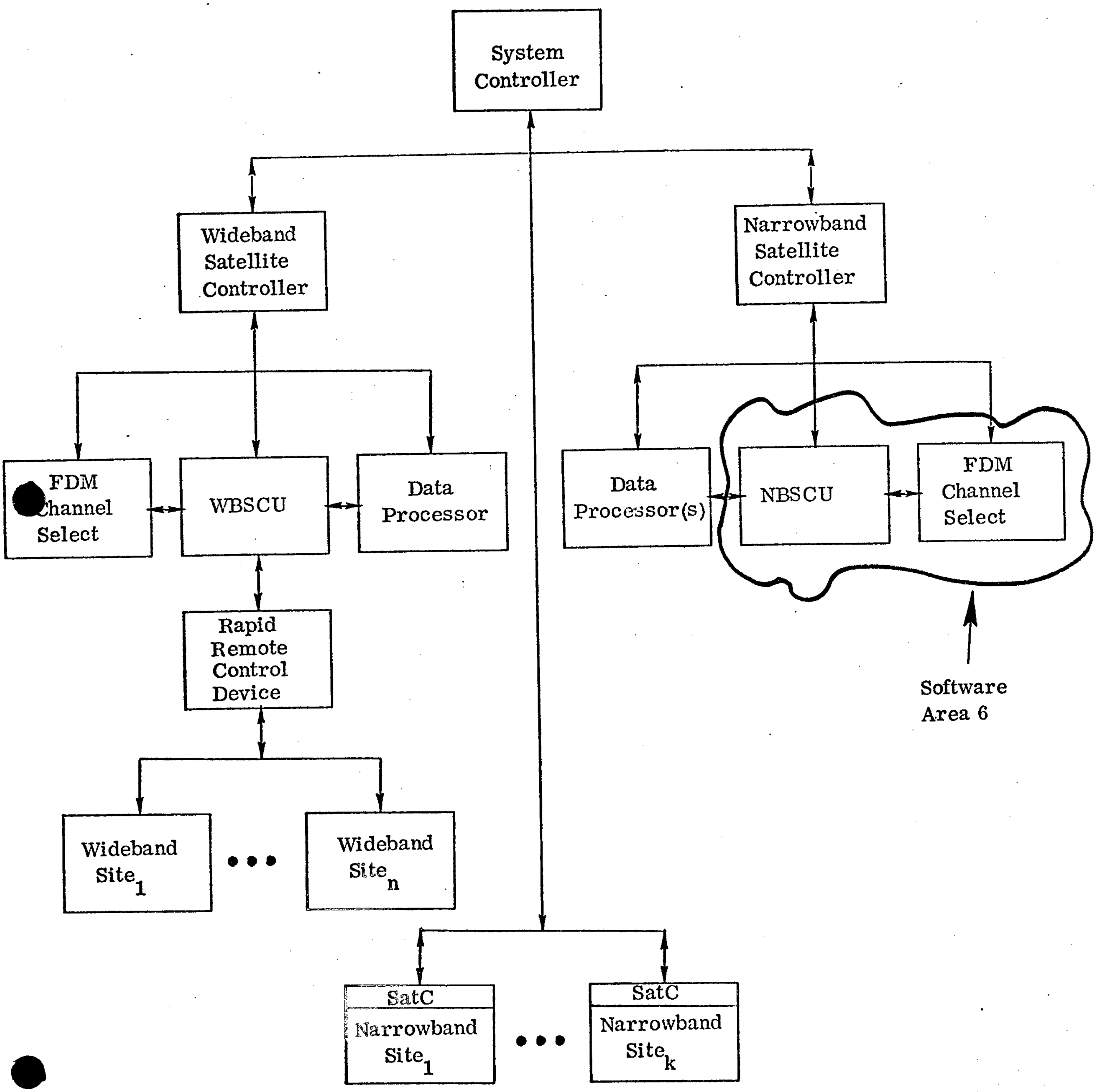


Table 1.3.1. Requirements for Software Area 1 - SysC

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION
Controller Executive	1.4.1.7	- Scheduling - Program Management - Subsystem Linkage - I/O Management
System Initialization (Configuration)	1.4.1.1	- When TIES is turned on or reset, the SysC will command each SatC to place the resources (Site) it controls into a predetermined configuration. - Preset each TIES resource in accordance with predefined parameters. - Works from information stored on media. - Automatically establishes a working system. - Does <u>not</u> activate RF emitters. - Modular enough to accommodate <u>any</u> complement of TIES resources installed in the given platform.
Reconfiguration	1.3.4.2.2	- The man-machine interface could allow definition of a new configuration after initialization.
Man-Machine Interface	1.3.4.2.1	- The need for a man machine interface arises from other requirements. - Change configuration parameters - Alerts - Status displays
Operator Intervention in Case of Failure	1.2.5 and Private Discussions	- Can be used to (partially) replace a failed SatC

Table 1.3.1. Requirements for Software Area 1 - SysC (Continued)

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION
Function Simulation	1.3.4.2.1	- Simulates non-existent hardware during TIES development.
Show and Tell	1.4.1.5	- Drive a display panel to demonstrate TIES capacities and techniques.
External Test Interface	1.4.1.6	- External Test could go through SysC interface.
Built-In Test (BIT)	1.4.1.2	- Run periodic tests through SatCs to insure TIES integrity. - Maintain a record of detected malfunctions. - Message wrap (Loop Back) test - Signal Level Tests - Narrow Band Loop back Tests
Graceful Degradation	1.4.1.3	- Error recovery for "planned" error situations. - Display warning and operator intervention modes for unplanned errors. - Crowbar mode for SysC computer or SysC software failures.
Inter-Computer Communication	1.3.4.2.2	- All control done through SatCs. This means that SysC must be able to talk to SatCs.

Table 1.3.2. Requirements for Software Area 2 - SatC

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION
Site Configuration Control	1.1.3.1	- Tunes resources under SatC control to settings specified by SysC.
Graceful Degredation	1.2.3	- Front-End bypass in case of front end failure. - Default configuration entered when SysC failure detected.
Built-In Test (BIT)	1.3.1.4	- SysC - initiated BIT. - Periodic BIT

Executive	Implied	- Device Interface along PLIB
		  - to varactors
		  - to synthesizers
		  - Automatic Gain Control
		  - Computer Controlled Gain
		  - to T/R switch
		  - to Front-end Bypass Logic
		- Message Interface along IEEE 488
		  - Message Encode
		  - Message Decode
		  - Message I/O
		  - ASCII - Binary
		  - Binary - ASCII
		- Interrupt Management
		- Scheduling

Table 1.3.3. Requirements for Software Area 3 - ExtT

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION

On-Line Tests	1.5.1	- Maintenance Test
		  - System Functions AT & E (test and prepare test report)
		  - Operator assisted Man-Machine Interface Tests.
		- Fault Localization to WRA level.

Off-Line Tests	1.5.1	- WRA Performance Tests (for repaired WRAs) - WRA repair program (to locate faulty SRAs)
SRA Repair and Test	Table 2	- This is a walk-through repair test program to be used in the repair of SRAs.
Test System Test	Table 2	- This will test the resources of the test system and provide go/no go status.

Table 1.3.4. Requirements for Software Area 4 - DatP

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION
Bit Formatting	1.1.3.3	
EDAC Coding/Decoding	1.1.3.3	
I/O Interface	1.1.3.3	

Table 1.3.5. Requirements for Software Area 5 - WBSCU

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION
Initialization	1.4.2.17	
Mode Selection	1.4.2.17	
Test Activation	1.4.2.17	
On & Off Bus Coupler Frequency Set & Reset	1.4.2.17	
Data Processor Software Loading	1.4.2.17	
Process JTIDS, TACAN, IFF/DABS, & GPS Waveforms	1.1.3.3	
Matched Filtering	1.1.3.3	Where S/W is involved
Pulse Detection	1.1.3.3	Where S/W is involved
Post Detection	1.1.3.3	Where S/W is involved
Transmit Premodulation	1.1.3.3	Where S/W is involved

Table 1.3.6. Requirements for Software Area 6 - NBSCU

REQUIREMENT DESCRIPTION	TSO & R REFERENCE	DISCUSSION
Modulate/Demodulate for any of a number of Modulation Types	1.1.3.3 & 1.2.3.2	- Examples include: - AM - FM - FSK - SSB - DCPSK
Low Rate Digital Voice Encode	1.1.3.3	
EDAC	1.1.3.3	
Graceful Degradation	1.2.3 & 1.2.3.2	- Should handle: - SysC Failure - SatC Failure - Failure of one NBSCU (interprocessor switch)
BIT	1.2.5	- Use one NBSCU as modulator. Use second NBSCU as demodulator. Switch roles.
Identical S/W for each NBSCU	1.2.6	
Handling all types of NBSCUs	1.3.3	- AM9080A - Proteus ASP - DEC LSI 11/03 & Mini Pro

SECTION 2. APPLICABLE DOCUMENTS

Documentation applicable to this specification is listed in Appendix A of this report. All abbreviations and terms with definitions are listed in a glossary in Appendix B.

SECTION 3. TACTICAL DIGITAL SYSTEM REQUIREMENTS

3.1 General

This section defines the functional, operational, and performance requirements that are necessary to insure the proper development and maintenance of TIES software.

3.2 Program Description

TIES is an integrated approach to communications, navigation, and identification requirements projected for diverse platforms. It's hardware differs from the hardware that would normally be found in more conventional, one-function-at-a-time systems oriented to the same requirements. TIES hardware is composed of general-purpose, usually programmable, units and is intended for multiband applications. The heart of the system is a broadband signal distribution system that allows any RF/IF signal path to be connected to any IF base band signal path. The basic system architecture consists of four subsystems - the control subsystem, the frequency conversion subsystem, the signal distribution subsystem, and the signal conversion subsystem.

The control subsystem consists of the SysC and the SatCs. They control TIES through SysC and SatC interfaces in a way that yields reconfigurability, capabilities for function simulation, potential for graceful degredation, and BIT. In addition, the control subsystem provides a framework for show and tell demonstrations.

The Frequency Conversion Subsystem acts to provide gain, filtering, and frequency conversion to signals received or emitted from TIES system. This subsystem consists of wideband and narrowband sites leading to antennas. Sites are controlled by satellite controllers (in the case of narrowband sites) or rapid remote control devices (in the case of wideband sites).

Table 3.2.1 lists those components that are significant to TIES software. Component names and station codes are tied to Figure 3.3.2 which is a block diagram of the area. The block diagram refers to narrowband UGF/VHF sites and the station codes should be considered in that light. BIT refers to the Built-in Test function.

The Signal Distribution Subsystem consists of two redundant pairs of FDM cables. Refer to Figure 3.3.1 for the relationship of the FDM to the remainder of the system.

TABLE 3.2.1 FREQUENCY CONVERSION SUBSYSTEM COMPONENTS SIGNIFICANT TO TIES SOFTWARE

COMPONENT	STATION (SEE FIGURE 3.3.3)	TYPE OF SITE WHERE ACTIVE: WIDEBAND	TYPE OF SITE WHERE ACTIVE: NARROWBAND	ACTIVE DURING: RECEPTION	ACTIVE DURING: TRANSMISSION	ACTIVE DURING: (RE) CONFIGURATION	ACTIVE DURING: BIT	DISCUSSION
Antenna Phasing & Gain Weighting	C	?	X	X				Phase Tuning
			X			X		Phase Tuning
UHF/VHF Select & Front-End Bypass	2		X			X		Translates VHF to UHF. Turned on if receiving VHF. Can be turned on in the case of front-end failure.
Varactor Filter	3		X			X		Varactor filter is set to received UHF frequency.
Power Supply	4	X	X	X	X			Turns power on and off: 1. On for transmit 2. On for receive 3. On for both 4. Off for both
IF Bandwidth Filter	5	X	X			X		Sets proper bandwidth by selecting appropriate filter.
IF Gain Characteristic	6	X	X			X		Sets IF to obtain desired gain characteristic and gain level (CCG).
Synthesizer Switch; Pilot Switch; Pilot Signal Bypass;	7	X	X	X				Injects Station F Signal to receive strip.
		X	X		X			Injects Station F Signal to transmit strip.
		X	X				X	Loop Testing.
Varactor Filter	8		X			X		Set to output frequency.
Power Amplifier; T/R Switch; Modulation Select; Attenuation Select;	A	X	X	X				Sets attenuator to high. Sets T/R to R.
		X	X		X			Sets power input level (3 Bits). Sets appropriate modulation (3 Bits). Sets T/R switch to T.
Varactor; Synthesizer; Bus Select Switch	D	X	X	X		X		Sets Varactor to the send FDM channel frequency. Selects the appropriate FDM bus (2-way switch). Sets synthesizer frequency to send FDM channel frequency.
Varactor; Synthesizer; Bus Select Switch	E	X	X		X			Sets varactor to the receive FDM channel frequency. Selects the appropriate FDM bus (2-way switch). Sets synthesizer frequency to receive FDM channel frequency.
Synthesizer	F	X	X	X				Sets the synthesizer frequency so that it will mix with the incoming signal to obtain the IF input.
		X	X		X			Sets the synthesizer frequency to obtain the proper transmit frequency.

The Signal Conversion Subsystem shown in Figure 1.3.7 consists of SatCs which control WBSCUs, NBSCUs, DatPs, and other devices. Table 3.2.2 lists those components that are significant to TIES software. Component names and station codes are tied to Figure 3.3.3 which is a block diagram of Software Area 6. The Block Diagram refers to narrowband equipments and station codes should be considered in that light. BIT refers to the Built-in Test function.

TABLE 3.2.2 SIGNAL CONVERSION SUBSYSTEM COMPONENTS THAT ARE SIGNIFICANT TO TIES

COMPONENT	STATION (SEE FIGURE 3.3.2)	TYPE OF UNIT WHERE ACTIVE		ACTIVE DURING				DISCUSSION
		WIDEBAND	NARROWBAND	RECEPTION	TRANSMISSION	(RE) CONFIGURATION	BIT	
FDM Channel Select	1 and 2	X	X			X		Selects bus and appropriate frequency.
Signal Conversion Units	3	X	X	X	X		X	NBSCU is configured for reception/ transmission.
Data Processor	4	X	X	X	X			DatP is configured for reception/ transmission.

3.2.1 General Description

The TIES system is an integrated approach to cost-effective C^3 NI platform transceiver capacity. The hardware approach is based on a standardized set of component modules that are designed for easy expansion and contraction to meet specific mission requirements. A variety of digital processor subsystems within the TIES milieu provide capacities for dynamic reconfigurative capacities, extended BIT, and graceful degredation. An extended digital system control capacity provides for a full set of man-machine interactive possibilities. The TIES ExtT system meets the maintenance requirements for the TIES CNI system. The ExtT system resources are able to localize the error, to perform communication system evaluation on a periodic basis, and to repair faulty WRAs after they have been removed from the aircraft.

Component performance specifications that further clarify performance requirements for TIES are listed below.

WIDE BAND SIGNAL CONVERSION UNIT (WBSCU) AND ASSOCIATED HARDWARE STATEMENT OF WORK NADC 4052, 25 October 1978, N62269-79-R-0013.

STATEMENT OF WORK FOR TIES NBSCU NADC 4052, 13 June 1978, N62269-79-R-0005.

TIES NARROW BAND SIGNAL CONVERTER General Dynamics, 1 August 1978.

FDM SUBSYSTEM SPEC

3.2.2 Peripheral Equipment Identification

TIES has several types of interfaces. For the purposes of this PPS, they are classified in terms of non-power and power interfaces.

3.2.2.1 Non-Power Interfaces

These interfaces constitute the peripherals by which people use TIES, They will include the AIDS System, CRTs, loudspeakers, microphones, teletypes, earphones, etc.

3.2.2.2 Power Interfaces

Power interfaces provide the energy to run TIES. They give standby redundancy.

3.3.3 Interface Identification

If TIES interfaces with other digital processor programs or tactical digital systems, it will do so through the data processors or the system controller. Since TIES is in a pre-ADM stage, these interfaces cannot be completely identified at this time. It is expected that AIDS will be one of these interfaces.

3.3 Functional Description

The major functions of the TIES software are documented in Tables 1.3.1-1.3.6 as amplified by Tables 3.2.1 and 3.2.2. The functional relationships of TIES with interfacing equipments and other digital processor programs are TBD.

3.3.1 Equipment Description

For the purposes of this document, the purposes of this document, the following equipments are considered to be part of the TIES system.

MAJOR UNITS

SysC

SatC

DatP

WBSCU

NBSCU

SITE COMPONENTS

Bus Convertors

Varactors

Synthesizers

Power Supplies

T/R Switch

IF Strip Controls

PLIB

UHF/VHF Convertors

Front End Bypass

MISCELLANEOUS

FDM Cable

RRCD (Rapid Remote Control Device)

External Test Equipment

They will be discussed in the following subparagraphs.

3.3.1.1 System Controller

The System Controller interfaces via the IEEE 488 to each Satellite Controller in TIES. Further, it interfaces with the human interface end of the TIES controller. The purpose of the System Controller is defined in Table 1.3.1 of this document. Equipment options and controls are TBS. These are documented for the Tektronix 4051 in references in Appendix A. Timing, resolution, and accuracy of the System Controller and associated interfacing equipment are TBS.

3.3.1.2 FDM Cable

The FDM cable is the medium by which signals are exchanged between the Frequency Conversion and Signal Conversion subsystems. Signals enter and are extracted via bus couplers. The frequency range of the FDM is 300-500 mhz. A 10 mhz channel with a 10 mhz guard band permits 20 simultaneous operative channels for each cable. Also, the FDM distributes an accurate 10 mhz frequency reference signal to all remote hardware.

3.3.1.3 Data Processors

The Data Processors are the interface between the signal conversion units and external system. They must be fast enough to handle the I/O requirements of these systems. The purpose of the equipment is documented in table 1.3.5 of this document. Equipment options and controls are TBS. Timing, resolution and accuracy requirements of the system are TBS.

3.3.1.4 Satellite Controllers

Satellite Controllers interface via the IEEE 488 Bus to the System Controller. They control Narrowband Sites, WBSCUs, and NBSCUs. Their general

functions are described in Table 1.3.2. Table 3.2.1 indicates the "peripherals" controlled by the Narrowband SatC. Table 3.2.2 indicates the peripherals controlled by Signal Conversion Unit (WBSCU and NBSCU) SatCs, timing, resolution, and accuracy of the SatCs and associated interfacing equipment are TBS.

3.3.1.5 Wideband Signal Conversion Units

WBSCUs interface with a SatC, one or more Data Processors, Bus Couplers, and a Rapid Remote Control Device which acts to control Wideband Sites. It is to be used to process JTIDS, TACAN, IFF/DABS, and GPS waveforms. The device is currently under procurement and timing, resolution, accuracy, and relevant interfaces remain TBS.

3.3.1.6 Narrowband Signal Conversion Units

NBSCUs interface with other NBSCUs, a Satellite Controller, one or more Data Processors, and Bus Couplers. They are used to process narrowband wave forms. Several classes of interim NBSCUs exist: a PROTEUS version, an AM9080 controlled version, and a PDP LSI 11/03 controlled version. Timing, resolution, accuracy, and much of the definitional work on the relevant control interfaces remain TBS.

3.3.1.7 Bus Convertor

Bus convertors are used to pick off and enter appropriate channels to and from the FDM. Channel selection is accomplished by computer controlled varactors and synthesizers. The varactors and synthesizers are controlled via SatCs. In

software terms, the Bus Convertors are indistinguishable from their associated varactors and synthesizers. One of two bus couplers is selected for use by a convertor by means of a bus select switch. Timing, resolution, and accuracy of the convertors are TBS.

3.3.1.8 Varactors

Digitally tuned varactor filters are used in a variety of locations in both Wideband and Narrowband sites and at on and off bus coupler sites in the Signal Conversion Subsystem. Their use is documented in Tables 3.2.1 and 3.2.2. Timing, resolution, and accuracy of the varactors are TBS. However, it is known that the number of tuning points to be selected differ among varactors.

3.3.1.9 Synthesizers

Digitally tuned synthesizers are used in a variety of locations in both Wideband and Narrowband Sites and at on and off Bus Coupler Sites in the Signal Conversion Subsystem. Their use is documented in Tables 3.2.1 and 3.2.2. Timing, resolution, and accuracy of the Synthesizers are TBS.

3.3.1.10 Power Supplies

A computer controlled power supply exists in each Wideband and Narrowband Site. It can condition the site for transmission only, reception, only both transmission and reception, or set the site off. This is documented in Table 3.2.1. Timing, resolution, and accuracy are TBS.

3.3.1.11 T/R Switch

A T/R (transmit/receive) switch exists in each Wideband and Narrowband Site. It allows routing signals between the site and the antenna for transmission or reception.

3.3.1.12 IF Strip Controls

IF strip controls provide gain and bandwidth limiting operations upon received signals at 70 mhz center frequency. Timing, resolution, and accuracy of the components are TBS.

3.3.1.13 Parallel Local Interface Bus

The Parallel Local Interface Bus (PLIB) is the transmission medium for control signals between the Satellite Controllers and the components they control. Specifications are TBS.

3.3.1.14 VHF/UHF Convertors

VHF/UHF Convertors exist in Narrowband Sites to translate received VHF signals to the UHF frequency band. The convertor is turned on if VHF is being received. Specifications for the device are TBS.

3.3.1.15 Rapid Remote Control Device

A Rapid Remote Control Device (RRCD) will exist in Wideband Sites to accommodate rapid turnaround requirements. The device is currently under investigation; its characteristics are TBS.

3.3.1.16 External Test System

The External Test System connects to TIES through the FDM subsystems and potentially through the DMDA, Antenna Couplers, and the System Resources Control Bus. The FDM connection allows the External Test System to pick

off or inject any IF from or to TIES. A similar situation applies to the DMDA link for baseband input and output signals. The Antenna Coupler allows signal injection and possibly output power measurement at the antenna terminal. The System Resources Control Bus connection allows simulation of human interface-type I/O with the System Controller.

3.3.2 Digital Processor Input/Output Utilization Table

TBS.

3.3.3 Digital Processor Interface Block Diagram

Several block diagrams are included in the following. The first (Figure 3.3.1) is a top-level block diagram for the entire system. The second (Figure 3.3.2) is a block diagram for a Narrowband Signal Conversion Site. The third (Figure 3.3.3) diagrams a UHF/VHF Narrowband Frequency Conversion Site. The final two diagrams (Figures 3.3.4 and 3.3.5) show Wideband Signal and Frequency Conversion Sites. The final two diagrams are reproduced from an RFP for a WBSCU that is under procurement.

3.3.4 Program Interfaces

TIES interfaces with external systems at the SysC and each DatP (along the DMDA).

The SysC interface (through AIDS) is the TIES human interface. It is used to (re) configure the system, access system status, and (potentially) initiate BIT. No details of the interface have been established.

The DatP interface exchanges messages with TIES and other systems on the host platform. No details of the interface have been established. Planned systems at this time include JTIDS, TACAN, IFF/DABS, GPS, NTDS, and various other communication systems.

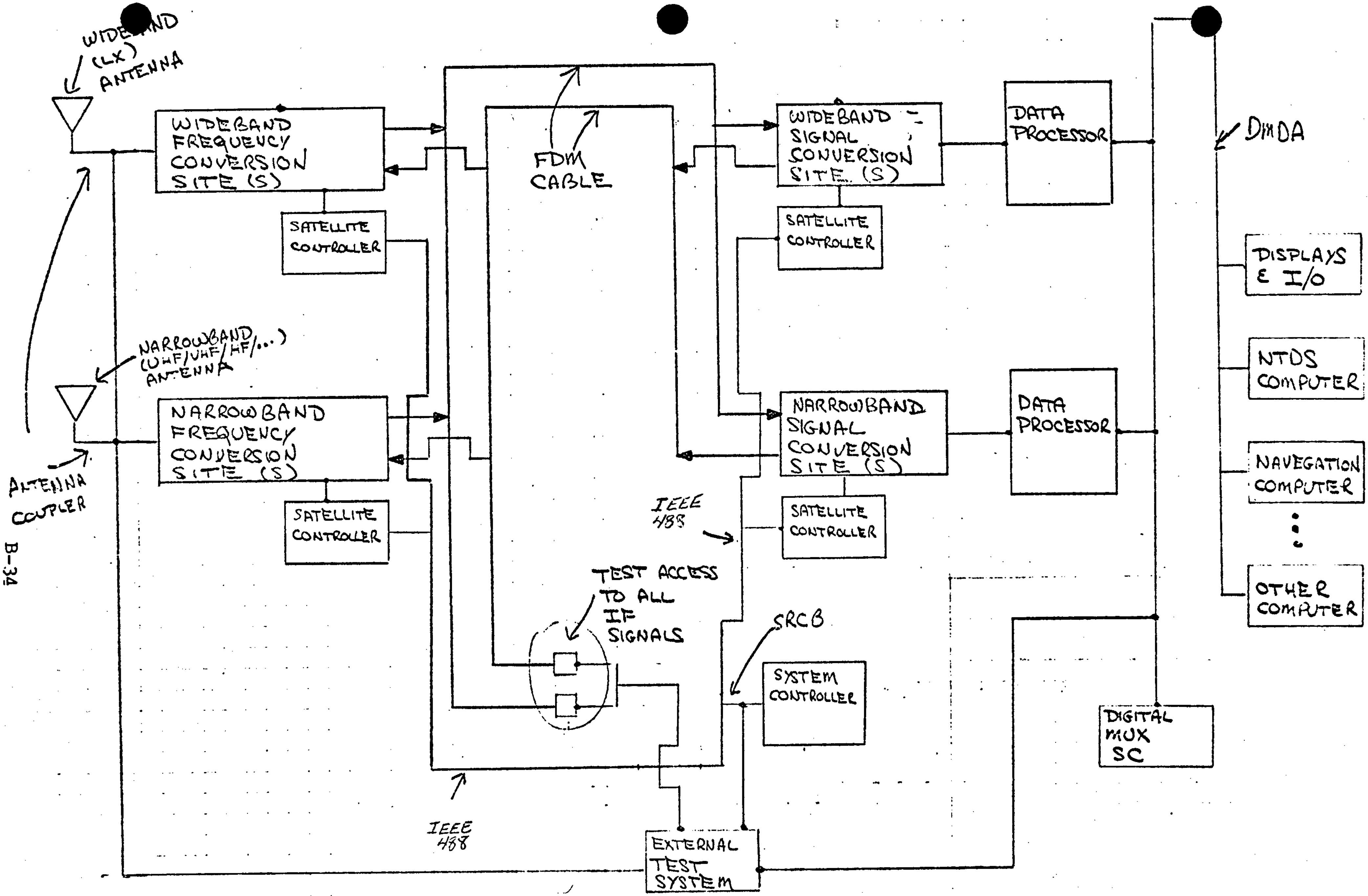


FIGURE 3.3.1 TIES BLOCK DIAGRAM (TOP-LEVEL)

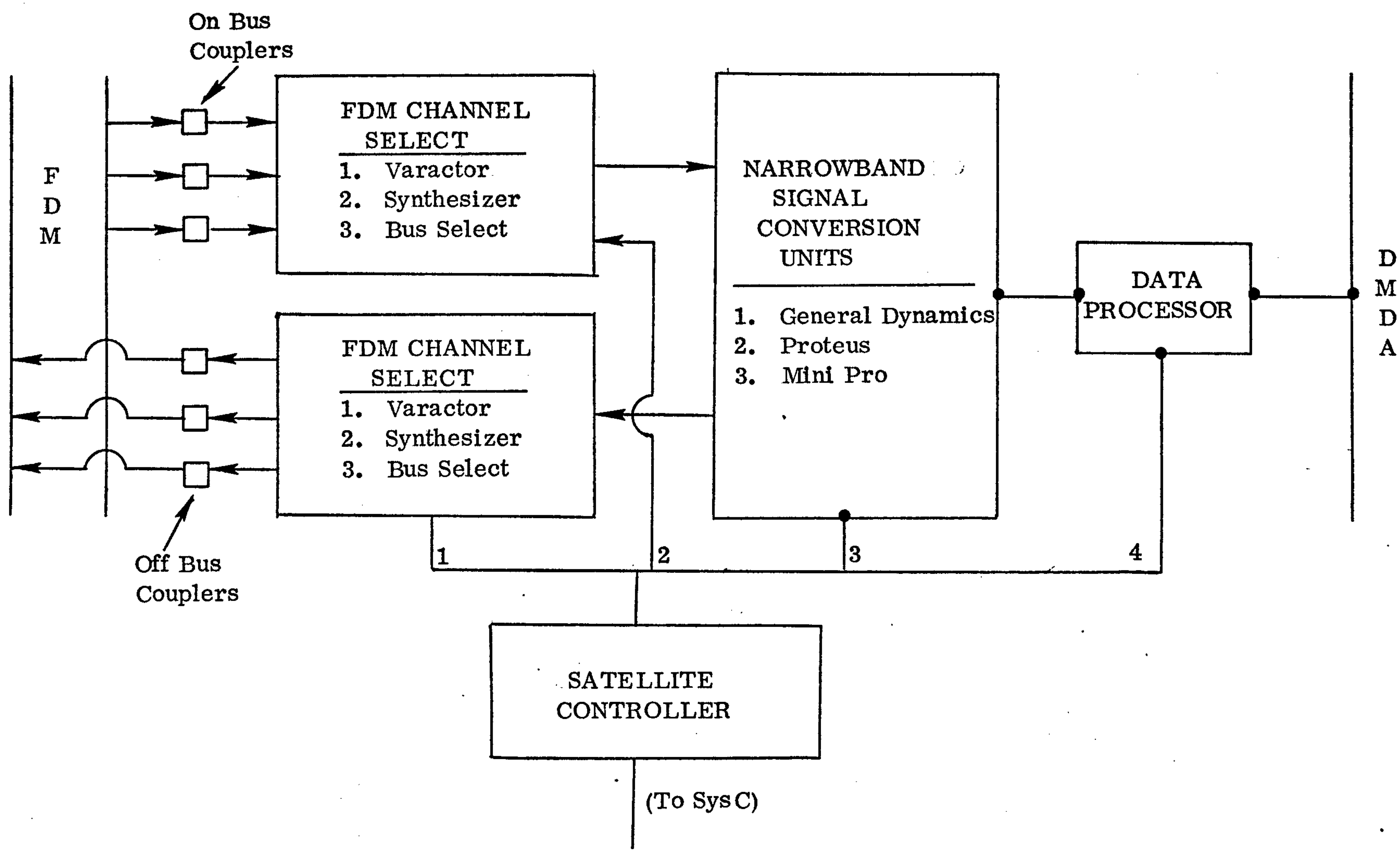


FIGURE 3.3.2 TIES BLOCK DIAGRAM FOR SOFTWARE AREA 6 (NBSCU) WITH ASSOCIATED DATA PROCESSOR AND SATELLITE CONTROLLER

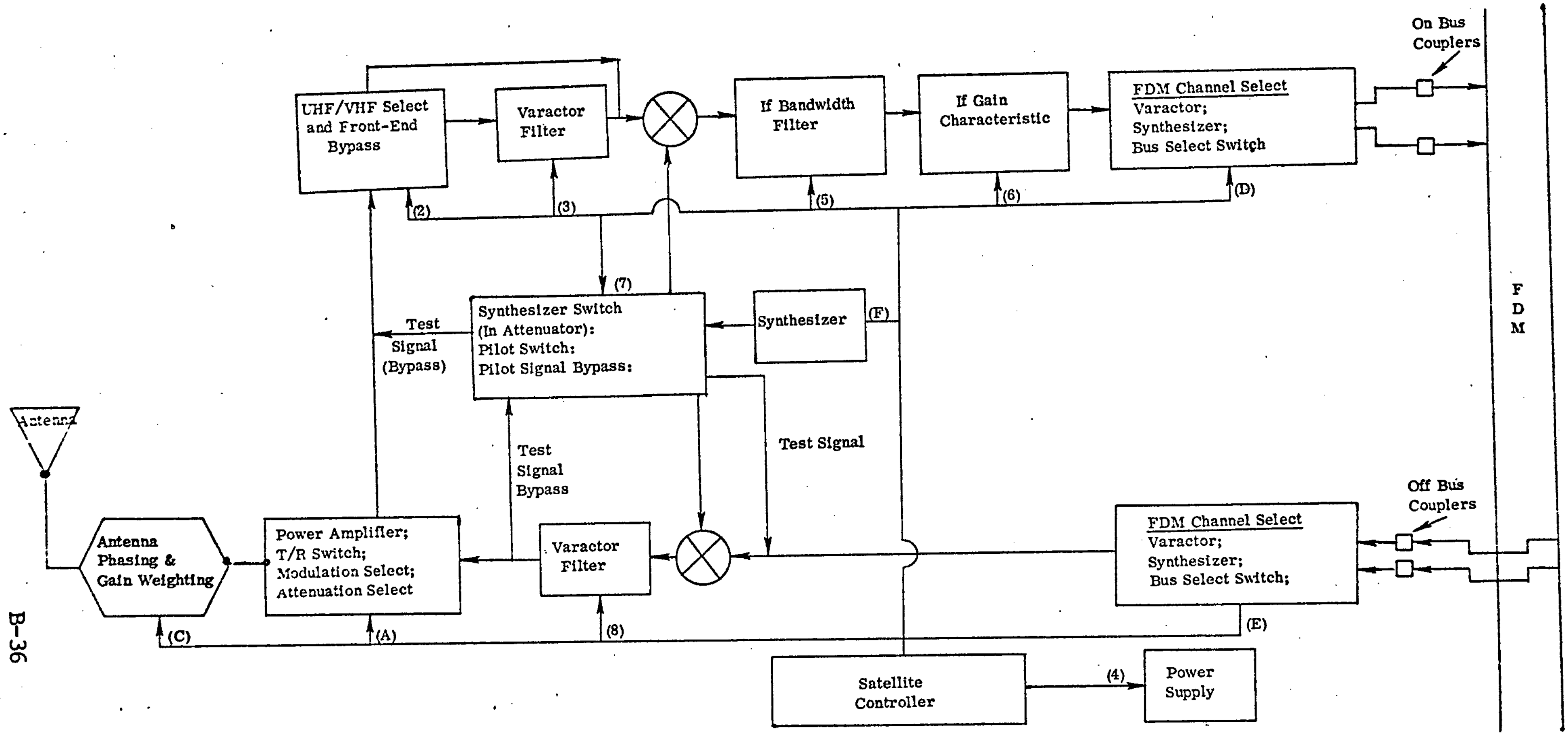
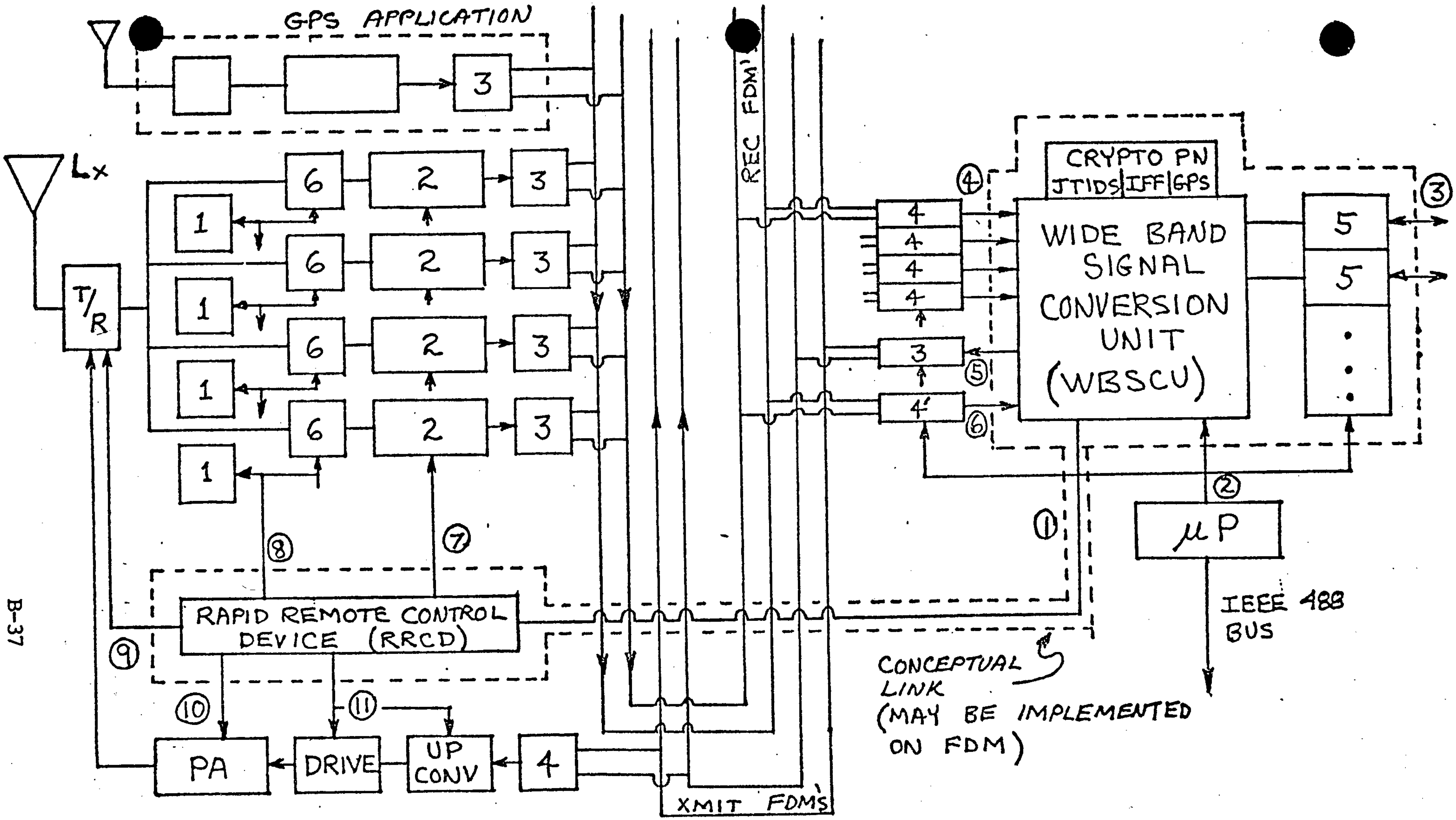


FIGURE 3.3.3 TIES BLOCK DIAGRAM FOR UHF/VHF NARROWBAND SITE



KEY:

Module List

1. Synthesizer
2. IF Strip
3. On Bus Coupler
4. Off Bus Coupler
5. Data Formatter
6. Receiver Front End

Interfaces to RFP List

1. RRC
2. PLIB
3. DMDA
4. FDM REC OFF BUS
5. FDM XMIT ON BUS
6. FDM CLOCK DROP
7. IF Control
8. Synth. and Rec Control
9. T/R Control
10. Envelop Control
11. Synth. Select

FIGURE 3.3.4 ASSOCIATED WIDEBAND FREQUENCY AND SIGNAL CONVERSION SITE

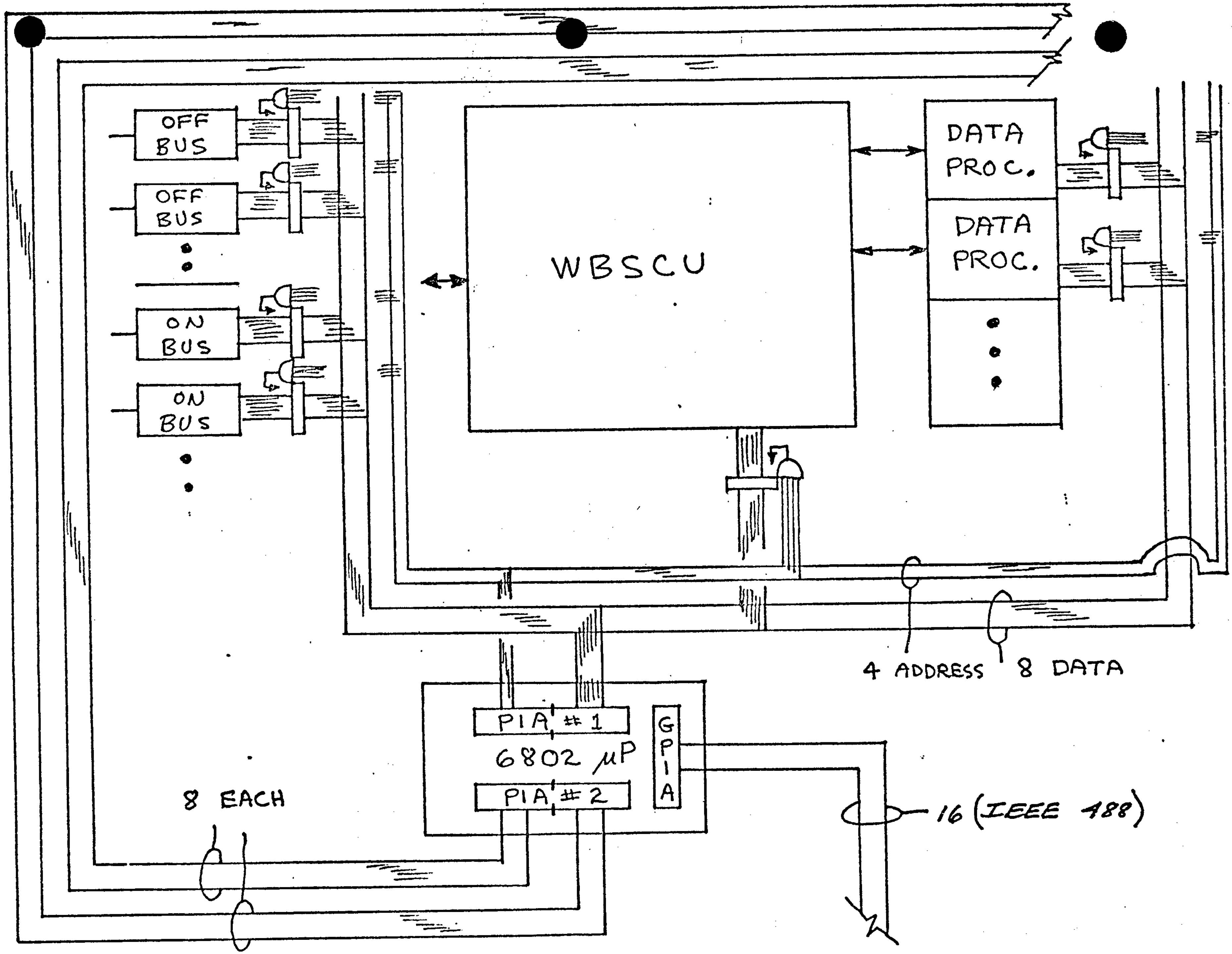


FIGURE 3.3.5. WIDEBAND SIGNAL CONVERSION SITE

3.3.5 Function Description

The following subparagraphs describe the functions to be supported by the TIES digital processor programs. There are several processor classes in TIES: each processor class implements different functional capabilities. Each processor class is discussed in its own subparagraph.

3.3.5.1 Functions Supported by the System Controller

3.3.5.1.1 The System Controller Data Base

The system controller (SC) data base has three major components:

- the RESOURCE TABLE,
- the ACTIVE MODE TABLE, and
- the MODE TABLE.

The RESOURCE TABLE is the first component of the Data Base. It describes the resources available to the particular TIES system. Each resource is described in terms of its location, its association with other resources (if any), its status, its association with a frequency or signal conversion unit, the transmit and receive capacity remaining, and the modes the resource can handle.

The ACTIVE MODE TABLE is the second component of the Data Base. This ACTIVE MODE TABLE exists to describe each mode the system is currently processing (or is to process). Each active mode is described in terms of its transmit and receive FDM channels. Further, since each mode requires a FCU and an SCU to do its work; the resources required to perform each function are identified by address. Finally, the parameters necessary to get the resource to perform the function are described.

The final component of the data base is the MODE TABLE. Modes will be encoded in memory using either a bit positional encoding or a prime encoding as is convenient in the host language. The encoding will be described in the MODE TABLE.

The following three tables describe mode encodings, resource encodings, and active mode encodings respectivly. The MODE ENCODINGS TABLE indicates that modes are catagorized by band and then are differentiated within band. In the example table, position 4 and the prime number 7 represent the mode JTIDS in the LX band, FSK in the UHF band, nothing in the UHF band, and TTY (FSK) in the HF band. The location of a resource in the RESOURCE ENCODING table is identified by its PRIMARY up ADDRESS and its SECTION ADDRESS. The SECOND up ADDRESS associates it with a second resource. The RESOURCE STATUS indicates whether the resource is UP (working), DOWN (not working), or NOT CHECKED YET (the TIES system hasn't yet been powered up). Remaining columns in this table and the TIES ACTIVE MODE ENCODINGS table are self explanatory.

TABLE 3.3.5.1 MODE ENCODINGS

NUMBER	CODE	LX	UHF	VHF	HF
1	2	DABS	AM VOICE	AM VOICE	AM
2	3	GPS	FM DATA	FM DATA	AM (SSB)
3	5	IFF	FM VOICE	FM VOICE	LINK 11
4	7	JTIDS	FSK		TTY/FSK
5	11	TACAN	LINK 4		

TABLE 3.3.5.2 RESOURCE ENCODINGS

COMPONENT	DESCRIPTION
Primary uP Address	Command Channel Address of Resource
Section Address	Section Number of Resource
Second uP Address	Command Channel Address of Associated Frequency/Signal Conversion Unit in Case of LX/WBSCU
Resource Status	UP/DOWN/NOT Checked Yet
Resource Type	Frequency/Signal Conversion Unit
Remaining Transmit Capacity	0 < = Remaining Transmit Capacity < = N
Remaining Receive Capacity	0 < = Remaining Receive Capacity < = 1
LX Modes Handled	Some Mode Encoding
UHF Modes Handled	Some Mode Encoding
VHF Modes Handled	Some Mode Encoding
HF Modes Handled	Some Mode Encoding

TABLE 3.3.5.3 TIES ACTIVE MODE ENCODINGS

COMPONENT		DESCRIPTION
Band		(LX, UHF, VHF, HF)
Mode Identifier		See Mode Encodings Table
Transmit FDM Channel		FDM Channel for Signals From SCU's to FCU's
Receive FDM Channel		FDM Channel for Signals from FCU's to SCU's
FCU Receive	uP Section Parameters	uP Address - or Don't Care Section Code - or Don't Care I/O Params
Transmit	uP Section Parameters	uP Address - or Don't Care Section Code - or Don't Care I/O Params
SCU Receive	uP Section Parameters	uP Address - or Don't Care Section Code - or Don't Care I/O Params
Transmit	uP Section Parameters	uP Address - or Don't Care Section Code - or Don't Care I/O Params

The following example encodes the resources for the TIES system that is diagrammed in the TIES lab as of 11 July 1979. Several assumptions have been made here to make the example more realistic.

The first assumption made is that the WBSCU consists of four sections the first of which can handle JTIDS transmission and reception while the remaining sections handle the transmission and reception of DABS, GPS, IFF, and TACAN. Note also how the WBSCU and the LX are tied together by the "SECOND uP ADDRESS" field of the description. The WBSCU is described in pages 9 through 12 of the example.

The second assumption is that the first NBSCU (controlled by uP 6) is a General Dynamics unit consisting of three sections each of which is capable of handling any UHF, VHF, or HF RF with the exception of LINK 4 or LINK 11.

The final assumption is that the second NBSCU is a Hughes unit with two sections each of which is capable of handling any UHF, VHF, OR HF RF.

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 001 LX - 1

COMPONENT	ENCODING
PRIMARY uP ADDRESS	1
SECTION ADDRESS	1
SECOND uP ADDRESS	5
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	0
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS/GPS/IFF/JTIDS/TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 002 LX - 2

COMPONENT	ENCODING
PRIMARY uP ADDRESS	1
SECTION ADDRESS	2
SECOND uP ADDRESS	5
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	0
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS/GPS/IFF/JTIDS/TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 003 LX - 3

COMPONENT	ENCODING
PRIMARY uP ADDRESS	1
SECTION ADDRESS	3
SECOND uP ADDRESS	5
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	0
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS/GPS/IFF/JTIDS/TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 004 LX - 4

COMPONENT	ENCODING
PRIMARY uP ADDRESS	1
SECTION ADDRESS	4
SECOND uP ADDRESS	5
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	0
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS/GPS/IFF/JTIDS/TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 005 LX - 5

COMPONENT	ENCODING
PRIMARY uP ADDRESS	1
SECTION ADDRESS	5
SECOND uP ADDRESS	5
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	10000
REMAINING RECEIVE CAPACITY	0
LX MODES HANDLED	DABS/GPS/IFF/JTIDS/TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES   PAGE 006 UHF/VHF FCU

COMPONENT	ENCODING
PRIMARY uP ADDRESS	2
SECTION ADDRESS	0
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	10000
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK,LINK 4
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES   PAGE 007 UHF/VHF FCU

COMPONENT	ENCODING
PRIMARY uP ADDRESS	3
SECTION ADDRESS	0
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	10000
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK,LINK 4
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES   PAGE 008 HF FCU

COMPONENT	ENCODING
PRIMARY uP ADDRESS	4
SECTION ADDRESS	0
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	FREQUENCY CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	10000
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	AM,AM <SSB>,LINK 11,TTY <FSK>

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 009 WBSCU - 1

COMPONENT	ENCODING
PRIMARY uP ADDRESS	5
SECTION ADDRESS	1
SECOND uP ADDRESS	1
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	JTIDS
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 010 WBSCU - 2

COMPONENT	ENCODING
PRIMARY uP ADDRESS	5
SECTION ADDRESS	2
SECOND uP ADDRESS	1
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS,GPS,IFF,TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 011 WBSCU - 3

COMPONENT	ENCODING
PRIMARY uP ADDRESS	5
SECTION ADDRESS	3
SECOND uP ADDRESS	1
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS,GPS,IFF,TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 012 WBSCU - 4

COMPONENT	ENCODING
PRIMARY uP ADDRESS	5
SECTION ADDRESS	4
SECOND uP ADDRESS	1
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	DABS,GPS,IFF,TACAN
UHF MODES HANDLED	
VHF MODES HANDLED	
HF MODES HANDLED	

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 013 NBSCU - GD(1)

COMPONENT	ENCODING
PRIMARY uP ADDRESS	6
SECTION ADDRESS	1
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	AM,AM <SSB>,TTY <FSK>

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 014 NBSCU - GD(2)

COMPONENT	ENCODING
PRIMARY uP ADDRESS	6
SECTION ADDRESS	2
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	AM,AM <SSB>,TTY <FSK>

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 015 NBSCU - GD(3)

COMPONENT	ENCODING
PRIMARY uP ADDRESS	6
SECTION ADDRESS	3
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	AM,AM <SSB>,TTY <FSK>

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 016 NBSCU - HUGHES

COMPONENT	ENCODING
PRIMARY uP ADDRESS	7
SECTION ADDRESS	1
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK,LINK4
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	AM,AM <SSB>,LINK 11,TTY <FSK>

EXAMPLE 3.3.5.1 TIES RESOURCES PAGE 017 NBSCU - HUGHES

COMPONENT	ENCODING
PRIMARY uP ADDRESS	7
SECTION ADDRESS	2
SECOND uP ADDRESS	0
RESOURCE STATUS	NOT CHECKED YET
RESOURCE TYPE	SIGNAL CONVERSION UNIT
REMAINING TRANSMIT CAPACITY	1
REMAINING RECEIVE CAPACITY	1
LX MODES HANDLED	
UHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE,FSK,LINK4
VHF MODES HANDLED	AM VOICE,FM DATA,FM VOICE
HF MODES HANDLED	AM,AM <SSB>,LINK 11,TTY <FSK>

The second example describes a hypothetical TIES MODE TABLE. Six modes are to be processed: JTIDS, IFF, TACAN, UHF AM VOICE, VHF FM DATA, and LINK 11. Each mode will be handled using resources described in the resource data base of EX 3.3.5.1. The first page of this example should read that JTIDS is in the LX band and uses FDM channels 1 and 2. Frequency conversion reception and transmission are handled by sections 1 and 5 of the LX unit. Signal conversion reception and transmission are handled by section 1 of the WBSCU. The remaining five pages of this example should be read in a similar way.

```
EXAMPLE 3.3.5.2 TIES MODES      PAGE 001
                ----------

COMPONENT                      ENCODING
---------                      ----------

BAND                           LX

MODE IDENTIFIER                JTIDS

TRANSMIT FDM CHANNEL           1

RECEIVE FDM CHANNEL            2

FCU  RECEIVE   uP              1
               SECTION         1
               PARAMETERS      PARAMS TO SETUP FCU RECEIVE

     TRANSMIT  uP              1
               SECTION         5
               PARAMETERS      PARAMS TO SETUP FCU TRANSMIT

SCU  RECEIVE   uP              5
               SECTION         1
               PARAMETERS      PARAMS TO SETUP SCU RECEIVE

     TRANSMIT  uP              5
               SECTION         1
               PARAMETERS      PARAMETERS TO SETUP SCU TRANSMIT
```

```
EXAMPLE 3.3.5.2 TIES MODES      PAGE 002
                ----------

COMPONENT                      ENCODING
---------                      ----------

BAND                           LX

MODE IDENTIFIER                IFF

TRANSMIT FDM CHANNEL           3

RECEIVE FDM CHANNEL            4

FCU  RECEIVE   uP              1
               SECTION         3
               PARAMETERS      PARAMS TO SETUP FCU RECEIVE

     TRANSMIT  uP              1
               SECTION         5
               PARAMETERS      PARAMS TO SETUP FCU TRANSMIT

SCU  RECEIVE   uP              5
               SECTION         2
               PARAMETERS      PARAMS TO SETUP SCU RECEIVE

     TRANSMIT  uP              5
               SECTION         2
               PARAMETERS      PARAMETERS TO SETUP SCU TRANSMIT
```

```
EXAMPLE 3.3.5.2 TIES MODES      PAGE 003
                ----------

COMPONENT                      ENCODING
---------                      --------

BAND                           LX

MODE IDENTIFIER                TACAN

TRANSMIT FDM CHANNEL           5

RECEIVE FDM CHANNEL            6

FCU  RECEIVE   uP              1
               SECTION         4
               PARAMETERS      PARAMS TO SETUP FCU RECEIVE

     TRANSMIT  uP              1
               SECTION         5
               PARAMETERS      PARAMS TO SETUP FCU TRANSMIT

SCU  RECEIVE   uP              5
               SECTION         3
               PARAMETERS      PARAMS TO SETUP SCU RECEIVE

     TRANSMIT  uP              5
               SECTION         3
               PARAMETERS      PARAMETERS TO SETUP SCU TRANSMIT
```

EXAMPLE 3.3.5.2 TIES MODES   PAGE 004

COMPONENT			ENCODING
BAND			UHF
MODE IDENTIFIER			AM VOICE
TRANSMIT FDM CHANNEL			7
RECEIVE FDM CHANNEL			8
FCU	RECEIVE	uP	2
		SECTION	0
		PARAMETERS	PARAMS TO SETUP FCU RECEIVE
	TRANSMIT	uP	2
		SECTION	0
		PARAMETERS	PARAMS TO SETUP FCU TRANSMIT
SCU	RECEIVE	uP	6
		SECTION	1
		PARAMETERS	PARAMS TO SETUP SCU RECEIVE
	TRANSMIT	uP	6
		SECTION	1
		PARAMETERS	PARAMETERS TO SETUP SCU TRANSMIT

EXAMPLE 3.3.5.2 TIES MODES   PAGE 005

COMPONENT			ENCODING
BAND			VHF
MODE IDENTIFIER			FM DATA
TRANSMIT FDM CHANNEL			9
RECEIVE FDM CHANNEL			10
FCU	RECEIVE	uP	3
		SECTION	0
		PARAMETERS	PARAMS TO SETUP FCU RECEIVE
	TRANSMIT	uP	3
		SECTION	0
		PARAMETERS	PARAMS TO SETUP FCU TRANSMIT
SCU	RECEIVE	uP	6
		SECTION	2
		PARAMETERS	PARAMS TO SETUP SCU RECEIVE
	TRANSMIT	uP	6
		SECTION	2
		PARAMETERS	PARAMETERS TO SETUP SCU TRANSMIT

EXAMPLE 3.3.5.2 TIES MODES   PAGE 006

COMPONENT			ENCODING
BAND			HF
MODE IDENTIFIER			LINK 11
TRANSMIT FDM CHANNEL			11
RECEIVE FDM CHANNEL			12
FCU	RECEIVE	uP	4
		SECTION	0
		PARAMETERS	PARAMS TO SETUP FCU RECEIVE
	TRANSMIT	uP	4
		SECTION	0
		PARAMETERS	PARAMS TO SETUP FCU TRANSMIT
SCU	RECEIVE	uP	7
		SECTION	1
		PARAMETERS	PARAMS TO SETUP SCU RECEIVE
	TRANSMIT	uP	7
		SECTION	2
		PARAMETERS	PARAMS TO SETUP SCU TRANSMIT

It would be inefficient to examine the resources of a TIES system in the resource-at-a-time formats of EX 3.3.5.1. For this reason, a Resource Table Display is defined. This display will show all the resources of a TIES system in a single display (or all that can be shown on one screen).

The following three pages describe the format of the display and show the resources of EX 3.3.5.1 formated in terms of the display.

FORMAT 3.3.5.1 RESOURCE TABLE DISPLAY FORMAT

ADDRESS	ST	F/S	RRC	RTC	T/R	BAND	MODE(S)
(uP,s,uP)	U	F	#	#	T	LX	VALID MODE(S) FOR THIS BAND
	D	S			R	UHF	
	?				""	VHF	
						HF	

NOTES:

1. "ADDRESS" COLUMN ENCODES:
   (PRIMARY uP ADDRESS,SECTION ADDRESS,SECOND uP ADDRESS)
2. "ST" COLUMN ENCODES RESOURCE STATUS INFORMATION
   U = UP
   D = DOWN
   ? = NOT CHECKED
3. "F/S" COLUMN ENCODES RESOURCE AS:
   F = FREQUENCY CONVERSION UNIT
   S = SIGNAL CONVERSION UNIT
4. "RRC" COLUMN ENCODES REMAINING RECEIVE CAPACITY
5. "RTC" COLUMN ENCODES REMAINING TRANSMIT CAPACITY
6. "T/R" COLUMN ENCODES CURRENT RESOURCE ACTIVITY
   T - TRANSMIT ACTIVITY (BAND AND MODE(S) COLUMNS DESCRIBE TRANSMIT ACTIVITY)
   R - RECEIVE ACTIVITY (BAND AND MODE(S) COLUMNS DESCRIBE RECEIVE ACTIVITY)
   "" - INACTIVE (RESOURCE IS UNUSED)

EXAMPLE 3.3.5.3 SAMPLE RESOURCE DISPLAY REPORT PAGE 1

ADDRESS	ST	F/S	RRC	RTC	T/R	BAND	MODE(S)		
(1,1,5)	?	F	0	0	T	LX	JTIDS		
(1,2,5)	?	F	1	0					
(1,3,5)	?	F	0	0	T	LX	IFF		
(1,4,5)	?	F	0	0	T	LX	TACAN		
(1,5,5)	?	F	0	96	R	LX	IFF	JTIDS	TACAN
(2,0,0)	?	F	0	98	T	UHF	AM VOICE		
					R	UHF	AM VOICE		
(3,0,0)	?	F	0	98	T	VHF	FM DATA		
					R	VHF	FM DATA		
(4,0,0)	?	F	1	99	T	HF	LINK 11		
					R	HF	LINK 11		

EXAMPLE 3.3.5.3 SAMPLE RESOURCE DISPLAY REPORT PAGE 2

ADDRESS	ST	F/S	RRC	RTC	T/R	BAND	MODE(S)
(5,1,1)	?	S	0	0	T	LX	JTIDS
					R	LX	JTIDS
(5,2,1)	?	S	0	0	T	LX	IFF
					R	LX	IFF
(5,3,1)	?	S	0	0	T	LX	TACAN
					R	LX	TACAN
(5,4,1)	?	S	1	1			
(6,1,0)	?	S	0	0	T	UHF	AM VOICE
					R	UHF	AM VOICE
(6,2,0)	?	S	0	0	T	VHF	FM DATA
					R	VHF	FM DATA
(6,3,0)	?	S	1	1			
(7,1,0)	?	S	0	1	R	HF	LINK 11
(7,2,0)	?	S	1	0	T	HF	LINK 11

It is also inefficient to examine the modes of a TIES system in the mode-at-a-time formats of EX 3.3.5.2. For this reason, a Mode Table Display is defined. The following three pages define this display and describe the modes of EX 3.3.5.2 in its format.

FORMAT .3.5.2 MODE TABLE DISPLAY FORMAT

```
                          S
                  R F T   E
                T E / / u C
BND MODE        R C S R P T PARAMETERS
--- ----        - - - - - - ----------
LX  MMMMMMMM    # # F R # # 1234567890123456789012345678901234567890
```

NOTES:
1. "BND" COLUMN SHOWS BAND ID
2. "MODE" COLUMN SHOWS MODE ID
3. "TR" COLUMN SHOWS FDM TRANSMIT CHANNEDL
4. "REC" COLUMN SHOWS FDM RECEIVE CHANNEL

5.
```
        S
F T     E
/ / u   C
S R P   T PARAMETERS
- - -   - ----------
F T {uP,SECTION,PARAMETERS} FOR FREQUENCY TRANSMIT
F R  "    "        "         "      "     RECEIVE
S T  "    "        "         " SIGNAL TRANSMIT
S R  "    "        "         "   "    RECEIVE
```

EXAMPLE 3.3.5.4 SAMPLE MODE DISPLAY REPORT PAGE 1

```
                          S
                  R F T   E
                T E / / u C
BND MODE        R C S R P T PARAMETERS
--- ----        - - - - - - ----------
LX  JTIDS       1 2 F T 1 1 PARAMS
                    F T 1 5 PARAMS
                    S R 5 1 1234567899012345678909112344765465647665
                    S T 5 1 PARAMETERS

LX  IFF         3 4 F R 1 3 LX IFF PARAMS
                    F T 1 5 FCU TRANSMIT PARAMS
                    S R 5 2 SCU RECEIVE PARAMS
                    S T 5 2 SCU RECIEVE PARAMS

LX  TACAN       5 6 F R 1 4 TACAN PARAMETERS
                    F T 1 5    "    "    "
                    S R 5 3  "    "    "
                    S T 5 3
```

EXAMPLE 3.3.5.4 SAMPLE MODE DISPLAY REPORT PAGE 2

```
                          S
                  R F T   E
                T E / / u C
BND MODE        R C S R P T PARAMETERS
--- ----        - - - - - - ----------
UHF AM VOICE    7 8 F R 2 0 PARAMETERS
                    F T 2 0   "
                    S R 6 1   "
                    S T 6 1     "

UHF FM DATA     9 A F R 3 0 ASDSLJASDFKJKLASFKLASFKLJKLLKFDFJKDFJFDS
                    F T 3 0
                    S T 6 2 STILL MORE PARAMETERS
                    S T 6 2

HF  LINK 11     B C F R 4 0 PARAMS PARAMS PARAMS
                    F T 4 0
                    S R 7 1
                    S T 7 2
```

3.3.5.1.2 TIES Initialization

SysC initialization takes place in five phases that are described below.

PHASE 1 PREINITIALIZATION

A description of the resources available to the TIES and any default settings are read from external media and the SysC data base described in 3.3.5.1 is constructed.

PHASE 2 DEFAULT ALLOCATION

Resource allocations defaulted in the preinitialization phase are performed by system allocation handlers. These handlers examine each mode to be processed. Each mode has resource requirements for frequency/signal transmission/reception. If a necessary resource assignment has not been made, the handler searches the resource table for an appropriate available device and makes the assignment.

PHASE 3 STATUS CHECKING

Resources are status checked (Do they exist? Do they work?).

PHASE 4 USER REVIEW

The user is given the opportunity to review the settings and change them if he wishes. Specifically

1. Resources can be made unavailable,
2. Resources or modes can be added or deleted, or
3. Resource assignments to modes can be changed.

PHASE 5 CONSISTENCY CHECK

The data base is checked for internal consistency. If inconsistency is detected, the user is informed and returned to phase 4 for corrections.

3.3.5.1.3 Configuration Specification

A working TIES system is configured when each site is given information about what it is to do. A TIES system would be in its initially configured or normal state when, for each mode the TIES is to process, for each resource assigned to the mode, if the resource is involved in signal reception - then the parameters to the resource have been sent by the SysC to the appropriate uP address for the applicable section.

3.3.5.1.4 Reconfiguration Specification

Reconfiguration involves the addition or deletion of a resource or mode or the reassignment of a resource to a mode. This can be done by entering phase 4 of 3.3.5.1.2. Mode addition with no resource assignment may require phase 2 of 3.3.5.1.2.

3.3.5.1.5 BIT Initiation and Analysis

The SysC will run interactive BIT on a periodic basis. The process essentially consists of looping test messages through sites and insuring that they are unchanged by the process. The SysC is required to inform the looping site that the test message is to be looped rather than to be handled in the normal manner. It is further required to inaugurate test message generation and analysis at the initiating site. Site failure would be handled as described in the last paragraph of 3.3.5.1.2.

An additional (non-interruptive) type of BIT exists in frequency conversion sites. This test loops a signal only through the site itself under SatC control. These tests can be made either independently of SysC control or at its command. Additionally, the SysC can request the results of the most recent non-interruptive test or be interrupted by the SatC to receive the information.

3.3.5.1.6 Crowbar Mode

If the SysC detects an internal failure within itself it will initiate a crowbar mode. This, in effect, informs each site that the SysC has failed and that each site should go to its backup mode.

FIGURE 3.3.5.1.5.1. BIT INITIATION AND ANALYSIS

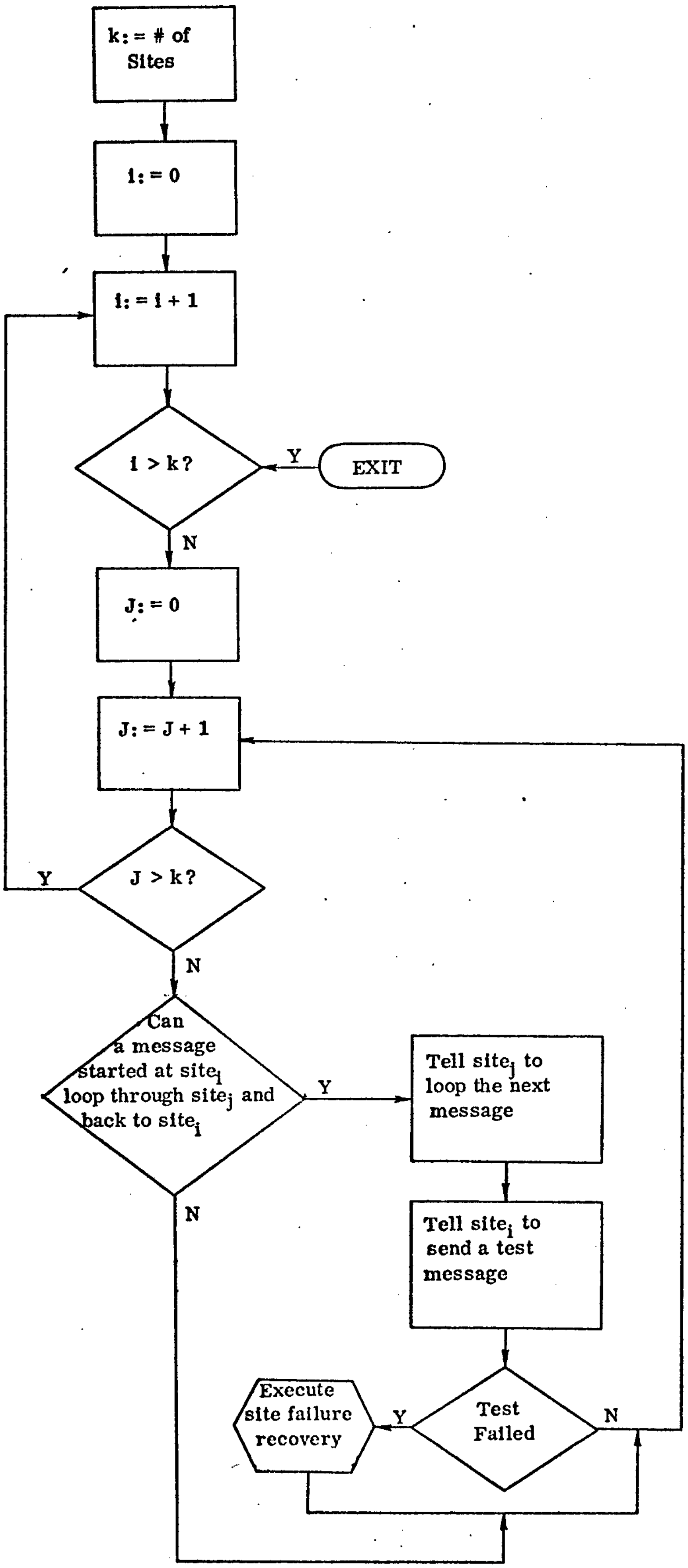
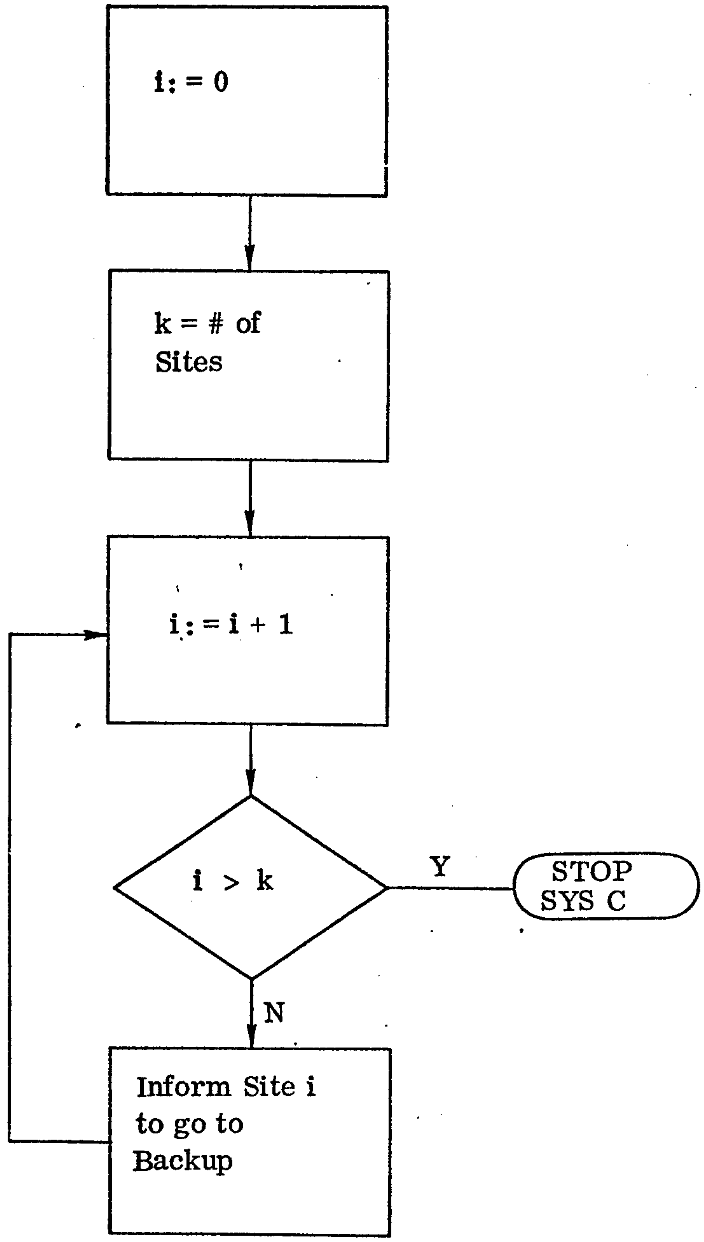


FIGURE 3.3.5.1.6.1. CROWBAR MODE



3.3.5.1.7 Function Simulation

The ADM TIES (and, perhaps, later versions) will have to be able to simulate non existant hardware. How this will be done is TBS.

3.3.5.1.8 Show and Tell

This function will be used in ADM TIES (and, perhaps, later versions) to drive a display panel to demonstrate TIES capacities and techniques. An early version of this function could simply keep an updated status display on the human interface. Function simulation will be used by show and tell.

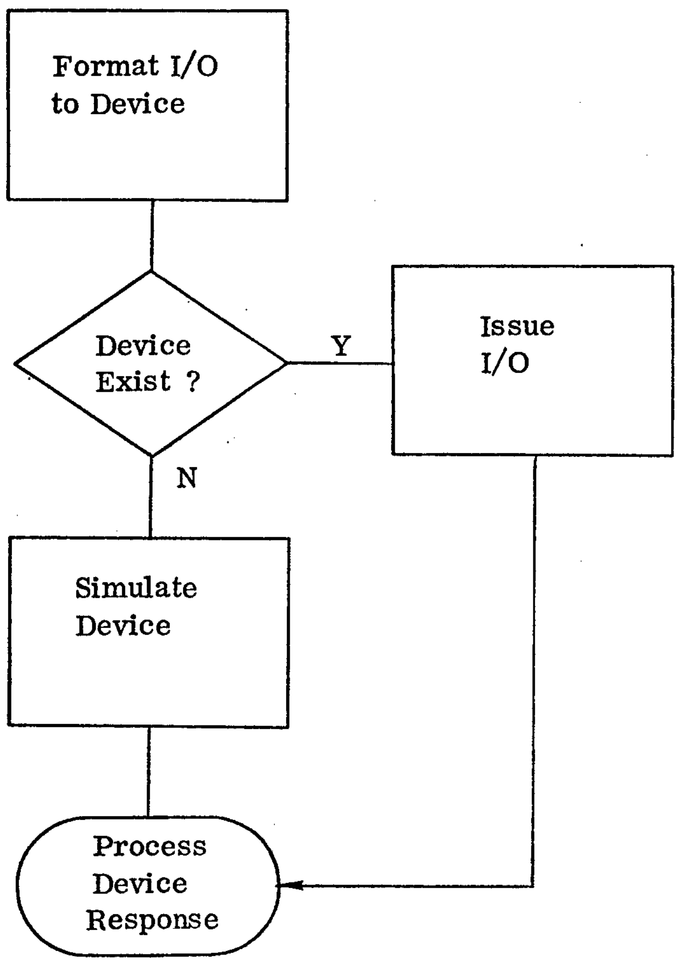
3.3.5.2 Functions Supported by the Satellite Controller

SatCs control Wideband Sites, Narrowband Sites, WBSCs, and NBSCs. Five general functions can be identified. They are:

- Reception
- Transmission
- Configuration/Reconfiguration
- BIT
- Graceful Degredation

Reception, transmission, configuration/reconfiguration, and BIT require adjustment of various components controlled by the SatC under discussion. Tables 3.2.1 and 3.2.2 discuss the component adjustments that are required for each of these four functions. However, these tables apply only to SatCs that control narrowband site or NBSCs. Wideband devices are under procurement and handling of each of these functions is still TBS.

FIGURE 3.3.5.1.6.1. FUNCTION SIMULATION



Three types of graceful degradation must be considered.

1. Recovery from SysC failure,
2. Recovery from SatC failure, and
3. Recovery from Component failure.

If the SatC is to recover from SysC failure, it must either

- Continuously operate as a transmitter if it is currently operating as a transmitter,
- Continuously operate as a receiver if it is currently operating as a receiver, or
- Go to its backup RF setup.

In order to handle its own failure, the SatC must inform the SysC of it's failure and stop it's current activities. The SysC will then compute a global backup posture.

If the SatC is to recover from component failure, how this is done depends on the failed component and is TBS.

3.3.5.3 Functions Supported by the ExtT

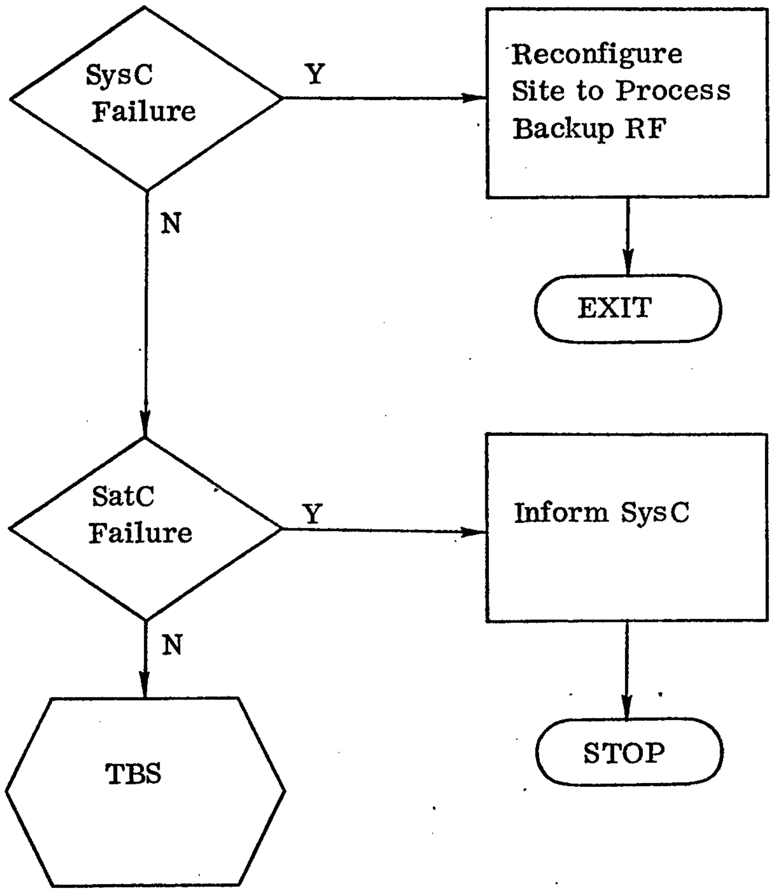
This material has been discussed previously in Table 1.3.3. It is amplified here.

3.3.5.3.1 On-Line Tests

3.3.5.3.1.1 Maintenance Tests

The Maintenance Test Programs will be responsible for checking all CNI System resources. These programs will evaluate receive and transmit functions of the frequency conversion subsystem equipment. It will also measure loss through the FDM subsystem, and locate possible discontinuities. The Maintenance

FIGURE 3.3.5.2.1. GRACEFUL DEGRADATION



program will determine the effectiveness of the frequency conversion subsystem, and it will exercise the various modes of the wideband and narrowband signal conversion units. Measurements to be made on the system include digital channel error rates, device frequency response, voice channel distortion measurements, FDM system insertion loss and the ability of the system controller to reconfigure the system upon receiving BIT sensor error signals. The BIT system elements shall be tested to make certain that they are operational. Upon completion of the system test, the system controller will exercise a BIT test routine to further insure proper operation.

The Maintenance program is divided into two sections. The first section is an automated testing routine which requires a minimum of operator assistance. This routine is used to determine system capability parameters by performing such tests as those mentioned above.

The second section is an operator assisted maintenance test. It will exercise the man-machine interface components of the TIES System. These components include microphones, headsets, CRT displays, and keyboards. The operator will be instructed by the external test controller to perform the tests at the various aircraft stations, and then will be given an allotted time to make the proper response. These functions must be tested but a detailed output is not required for the communication system status report.

3.3.5.3.1.2 Fault Localization

The Fault Localization program will be responsible for determining where a fault exists down to the WRA level. The ability to localize down to one WRA is desirable. The connections required to connect the aircraft to the external test system for the Fault Localization Program will be identical to those required for the Maintenance Program. The operator will be required to aid in the test of the aircraft system, however, this interaction should be minimal. Upon replacing the faulty WRA,

the Maintenance Test Program should be performed to insure that the repairs have been implemented properly.

During on-line testing, the aircraft to be tested would be connected to the external test system through FDM, DMDA, and computer control bus cables, as well as the test systems antenna couplers. The amount of work performed by the operator should be minimal, and the test program should explicitly tell the operator what connections to make, and what other inputs are needed to conduct the test.

3.3.5.3.2 Off-Line Tests

Off-line test routines include the WRA-Test and WRA Repair Programs. The WRA-Test Program would be used to check an assembly which has been repaired before it is installed on the aircraft. The WRA Repair Program is a walkthrough routine designed to locate faulty SRAs. The testing of WRAs will be aided by the use of common connectors on all WRAs.

The WRA test program will be used to test a WRA which has just been repaired to insure that it is functioning properly. These routines should output only the status of the device, with a selectable option to print out the measured specifications.

The WRA Repair programs are off-line tests designed to pinpoint WRA problems to the SRA level. These tests are conducted with the WRA installed on the test stand. The WRA tests may require operator actions in order to pinpoint errors, however, the program should be capable of localizing any problems to several possibilities automatically. After the WRA has been repaired, the WRA should be tested with the WRA Test Program. This will insure that the unit is operating properly when it is installed on the aircraft.

3.3.5.3.3 SRA Test and Repair

SRA Repair Programs will be written for a unit repair capability. The testing of SRAs will be aided by the use of a limited number of connector types on SRA units. This may be possible by controlling the SRA function with a local microprocessor which would communicate to the SRA through the PLIB (Parallel Local Interface Bus).

In order to repair SRAs at the aircraft facility, walkthrough SRA repair test programs will be developed. This test routine may require a large amount of operator intervention, depending upon the function and complexity of the SRA involved.

3.3.5.3.4 Test System Test

The ExtT System will have a test program to test the resources of the test system. This routine will provide a GO/NO-GO status after being executed, and if a NO-GO condition exists, the program should be capable of localizing the errors to the device level.

3.3.5.4 Functions Supported by the DatP

The functions supported by the DatP are those that are necessary to support the requirements described in Table 1.3.4. Not enough is known to specify their form and implementation at this time.

3.3.5.5 Functions Supported by the WBSCU

The functions supported by the WBSC are those that are necessary to support the requirements described in Table 1.3.5. Not enough is known to specify their form and implementation at this time.

FIGURE 3.3.5.3.1.1. ON-LINE TESTS

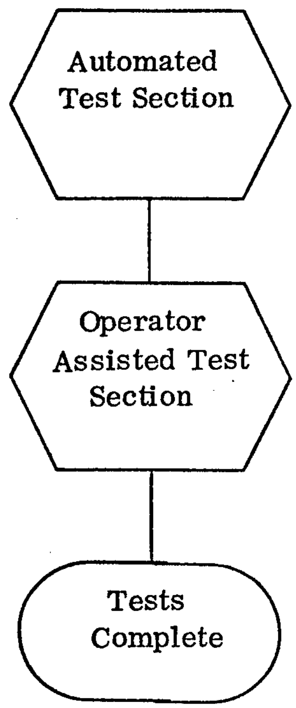


FIGURE 3.3.5.3.1.1.1. AUTOMATED TEST SECTION

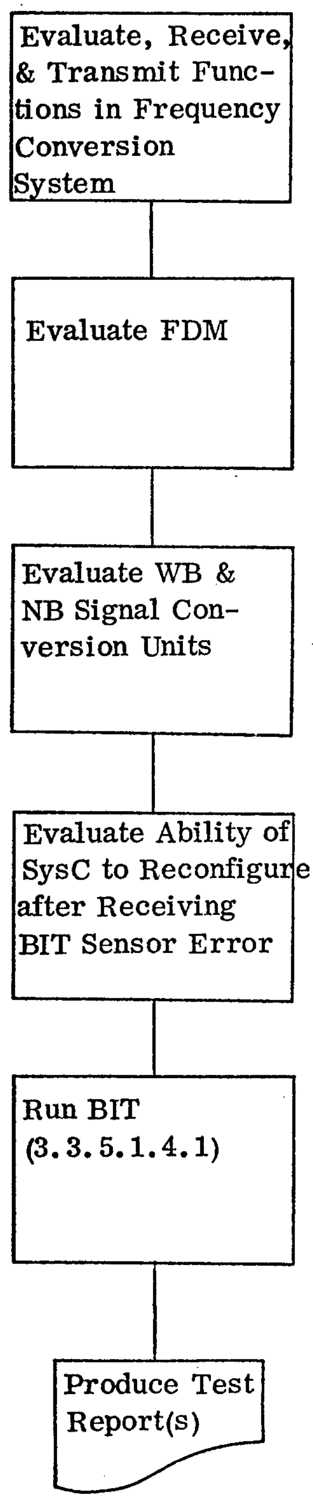


FIGURE 3.3.5.3.1.1.2. OPERATOR ASSISTED TEST SECTION

Allow an Operator the Capacity to Test all Man-Machine Interfaces with TIES

3.3.5.6 Functions Supported by the NBSCU

The functions supported by the NBSC are those that are necessary to support the requirements described in Table 1.3.6. Not enough is known to specify their form and implementation at this time.

3.4 Detailed Function Requirements

3.4.1 Resource Determination

Not enough is known to extend the description of resource determination beyond the discussion in 3.3.5.1.1.

3.4.1.1 Inputs

See examples 3.3.5.1 and 3.3.5.2.

3.4.1.2 Processing

Not enough is known to extend the textual description of Resource Determination beyond that described in 3.3.5.1.1.

The function serves two purposes. The first purpose is to verify whether the system, as it exists, is capable of being configured in some way. For example, if only two Narrowband Frequency Conversion Sites exist, the SysC could determine that an attempt to run three simultaneous narrowband channels would be impossible. The second purpose is to allow the SysC to determine if alternate sites exist in the event of site failure.

The function has no parameters.

3.4.1.3 Outputs

The output of the resource determination function is the SysC data base. Each site entered is checked to insure that it is, in fact, operative. If it is, the site description is entered into the SysC data base. If it is not, the operator is informed that the site is currently down. Resource determination should not fail simply because one of the sites is not operative.

3.4.2 Configuration Specification

Not enough is known about configuration specification to extend the description beyond 3.3.5.1.3.

3.4.2.1 Inputs

The function's inputs are described in 3.3.5.1.2.

3.4.2.2 Processing

Not enough is known to extend the textual description of configuration specification beyond 3.3.5.1.3.

3.4.2.3 Outputs

See examples 3.3.5.3 and 3.3.5.4.

3.4.3 Reconfiguration Specification

A detailed description of reconfiguration requires a more detailed description of configuration specification. Hence, not enough is known to extend the description beyond that of 3.3.5.1.4.

3.4.3.1 Inputs

See example 3.3.5.1.4.

3.4.3.2 Processing

See 3.3.5.1.4.

3.4.3.3 Outputs

See 3.3.5.1.4.

3.4.4 BIT Initiation and Analysis

The goal of a BIT cycle is to create a performance matrix as described in Figure 3.4.4. The matrix can be interpreted in the following way.

- A row containing no S indicates the site is probably failing to generate messages properly.
- A column containing no S indicates the site is probably failing to receive messages properly. (However, if the equivalent row contained no S this may indicate only generation failure.) Note, that if the message received at the looping site is intercepted and routed to the SysC, a more accurate analysis of column errors can be made.

FIGURE 3.4.4. BIT PERFORMANCE MATRIX

Loop Site / Initiate Site	$Site_1$	$Site_2$	$Site_3$	o o o	$Site_n$
$Site_1$		S			
$Site_2$			F		
$Site_3$					
o o o o					
$Site_n$	F				S

NOTES

1. Rows denote initiating sites
2. Columns denote looping sites
3. Column entries are

   Blank - No BIT Test Possible

   S - BIT Message Loop Successful

   F - BIT Message Loop Failed

4.4.1 Inputs

Inputs are test messages sent from the SysC to the site SatCs.

3.4.4.2 Processing

See paragraph 3.3.5.1.5 The purpose of the function is to determine if failed sites exist so that the system can be reconfigured. There are no parameters to the function.

3.4.4.3 Outputs

The output of the BIT function is the test matrix described in Figure 3.4.3. Failed sites are handled by reconfiguring the TIES.

3.4.5 Crowbar Mode

SysC failure is a most difficult issue. Two kinds of failure must be examined. The first is "planned" or "manageable" failure. Typical examples of this include software divide by zero and hardware parity error. Unplanned errors include loops in the software, hardware errors, I/O errors, coding errors, and so on.

3.4.5.1 Inputs

Not applicable.

3.4.5.2 Processing

Planned failures should be handled by having the SysC immediately notify all sites of its failure.

Unplanned failures can be (somewhat) managed by having each site require the SysC to turn around some test message once every period of time (if no transmissions from the SysC have been received during that period). Failure to perform this action properly would be evidence of SysC failure. (Note, however, that a successful turnaround would not guarantee that the SysC was operating properly.)

3.4.5.3 Outputs

Not applicable.

3.4.6 Function Simulation

The goal of function simulation is to simulate the performance of non-existent hardware components. The approach is reasonable whenever interactions with the non-existent device are performed in a systematic manner. Table 3.4.6 indicates the possibilities.

3.4.6.1 Inputs

Inputs to the simulations come in two forms. The first is the normal application-level communicate to the device through the software executive. The second input is a clock-driven or scenario driven interrupt which signals a simulation-level input from the non-existent device.

3.4.6.2 Processing

Commands to the device are analyzed by the executive and passed to the appropriate simulation routine for handling. Handling will be device-specific. Outputs from the simulated device are generated by the device simulation code, formatted by the executive, and passed to the applications level software.

TABLE 3.4.6 OPPORTUNITIES FOR FUNCTION SIMULATION

DEVICE TO BE SIMULATED	PROCESSOR(S) REQUIRED	SCENARIO CONTROL?	FUNCTIONS SIMULATED
Wideband Signal Conversion Site	SysC and Wideband Frequency Conversion Site	Possible	Any Normal Wideband Site Function
Narrowband Signal Conversion Site	SysC	Possible	Any Normal Narrowband Site Function
Wideband Frequency Conversion Site	SysC and Wideband Signal Conversion Site	Possible	Any Normal Wideband Site Function
Narrowband Frequency Conversion Site	SysC	Possible	Any Normal Narrowband Site Function
DatP	(1) SatC (2) SysC and SatC	(1) Not Possible (2) Possible	Message I/O & Message Analysis
Any Site Specific Hardware	SatC	Not Possible	Any Normal Hardware Function

3.4.6.3 Outputs

Not applicable.

3.4.7 Show and Tell

Show and Tell will drive a large display panel that will demonstrate TIES functions, capacities, and techniques. Some of the things to be demonstrated are listed in Table 3.4.6.

3.4.7.1 Inputs

See table 3.4.7.

3.4.7.2 Processing

TBD. This will be application-specific and undergo continual modification.

3.4.7.3 Outputs

See table 3.4.7.

TABLE 3.4.7 ANTICIPATED SHOW AND TELL CAPABILITIES

FUNCTION/CAPACITY	INPUTS	OUTPUTS
TIES Schemetic	# and Types of Sites	Schemetic of TIES System With Sites Requested
Signal Path Demo	Point of Signal Origination and Point of Signal Termination	Show Signal Paths through TIES System
Site Failure Demo	Which Site(s) Failed	Indicates which Sites (if any) Handle Failed Site's Functions
Site Component Failure Demo	Which Component(s) Failed in which Site	Shows how Site would Perform Degraded Mode Operations
SysC Failure Demo	None	Shows how Sites got back-up Configurations
BIT Demo	None	Demonstrates BIT
Receive/Transmit Demo	None	Indicates Control flow in Handling Signal Reception or Transmission

SECTION 4. QUALITY ASSURANCE

4.1 General

Software quality assurance (QA) is designed to assure accurate and sufficient planning, controlling, and reporting to affect the development of software products that meet their requirements.

The functions of software QA should be performed by an organization reporting to the project manager. The QA function has the control, responsibility, and authority for these eight basic functions:

1. Initial quality planning
2. Development of software standards and procedures
3. Development of quality assurance tools
4. Conduct of QA reviews and audits
5. Inspection and surveillance of formal tests
6. Configuration verifications
7. Management of the discrepancy reporting system
8. Retention of QA records.

In the following subsections, each function will be discussed. Detailed QA procedures can be found in applicable division QA manuals.

4.1.1 Initial Quality Planning

Successful implementation of the QA program for TIES depends heavily on QA planning during the early phases of the software development life cycle. This is accomplished by a complete review of project documentation. The review has cumulated in the preparation of a QA Plan given below that contains the quality assurance functions, tasks, responsibilities, and identifies the QA tools needed to assure sufficient software quality with regard to accountability, testability, usability, maintainability, and reliability.

When task assignments are made to carry out QA functions, these assignments are usually based on level of effort and must remain flexible to adapt to:

1. The needs of the current phase in the development life cycle
2. Shifts in areas needing attention (e.g., technical problems)
3. Unscheduled demands placed on QA by the project manager.

After QA Plan approval, QA policies and procedures must be written to describe the methods and procedures to be used in implementing the quality assurance requirements as they will eventually be defined in subsequent specifications.

The benefits of keeping the QA up-to-date, setting priorities, and maintaining project-wide visibility cannot be underestimated. QA effectiveness is directly related to its degree of involvement in project planning and to its visibility into the development of software products.

4.1.2 Development of Software Standards

Software standards are designed to improve the maintainability and readability of the software product. For all software projects, a comprehensive and detailed software standards program should be developed and implemented to fulfill this purpose. The effectiveness of a software standards program can be increased by two factors:

1. Software standards must be developed out of close communication among the design and development, test, QA, and project offices.
2. A tool should be provided to automatically check the software against most of the standards. This will allow programmers to audit themselves so that there will be no surprises at turnover time.

Waivers and deviations should complement the standards by providing a mechanism for management approvals of non-conformance with standards in particular situations. Waivers and deviations provide relief from standards compliance due to, for example, technical difficulties, inefficiencies, or schedule impact, and must be approved, at a minimum, by both the QA and project manager.

4.1.3 Development of QA Tools - FASP

QA tools are essential to the successful implementation of a software QA program because they provide a cost effective method for accurately evaluating large software products in a short period of time. Generally, QA tools should perform tasks that are more cost effective and accurate if done automatically than if done manually. Tools may be used to support any QA task, but experience has shown their high payoff in the following areas:

1.	Code Auditing -	Tools used to read source code for compliance to a prespecified set of programming standards.
2.	Test Monitoring -	Tools used during testing to measure test coverage, i.e., what code is tested by each test case, etc.
3.	Requirements Validation -	Tools used to track and report the test level and test procedure that validate each requirement.
4.	Configuration Verification -	Tools used to assure proper library control by reporting all changes to source code files.
5.	Discrepancy Reporting -	Data base management systems set up to log, track, and report the status of all project discrepancy reports.

FASP, the Facility for Automated Software Production, is a program generation facility in which operational and system test (diagnostic) software can be developed and maintained. Because of its standardized and automated nature, FASP reduces software development and maintenance costs.

FASP provides a uniform software development and maintenance environment.

a. FASP moves the development and maintenance activity from the target machine to large scale commercial computers where programming support tools are available.

b. FASP provides for continuity between development and maintenance.

c. FASP makes delivery of the software to the Navy or to any other group easy.

FASP provides a basic fonfiguration management function.

a. FASP provides for the synchronization of source code, object code and modification information.

b. FASP automatically maintains modification information and an audit trail.

c. FASP maintains simulator test data and hardware load tape creation directives.

FASP supports the project programmer.

a. The FASP command language provides programmer access to all required program design, generation, test, reporting and delivery functions. The command language is function-oriented and easier to learn and use than the Control Data Corporation KRONOS operating system.

b. FASP automatically keeps records that the programmer would have to keep manually under other systems: Modification Info. Table, Comdeck Index Table, External Entry Table, Software Management Reports.

c. FASP provides automatic mass storage backup for the data base and makes backup to magnetic tape easy.

d. FASP provides for testing using interpretive language simulators that do not require access to the target hardware.

FASP consists of a set of procedures that run on the NAVAIRDEVCEN CCS (Central Computer System) and manipulate a data base which contains the users' software. The NAVAIRDEVCEN CCS is a large complex of third generation commercial computers, including one CDC CYBER 175 and two CDC 6600's, plus extensive mass

memory and supporting peripheral devices. Located at NAVAIRDEVCEN, the CCS can support remote operations either through terminals on a multishift basis or through computer-to-computer links.

The FASP data base is automatically maintained on permanent file on the NAVAIRDEVCEN CCS. The integrity of the data base is maintained by using an on-line backup. The following information is contained in the FASP task data base:

a. Software program libraries. A consistant set of source and object libraries.

b. Program development and modification information. Descriptive information required by programmers and management that pertains to the developed software.

c. Load and test information. The information necessary to generate load modules, and perform and analyze tests of the developed software.

d. Data base management information. The information required by FASP to maintain the data base.

FASP manipulates the data base with a set of KRONOS procedures that are the users' interface to the data base. Each FASP procedure (or command) describes a particular programmer task as a logical function. Typically, each task is a multistep sequence of operations that FASP performs as the result of a one-line command. Simple user-oriented FASP commands are provided to perform the following software development functions:

a. Assemble/compile/translate source program
b. Maintain source and object libraries
c. Provide for data base backup and automated recovery
d. Perform software testing and validation
e. Maintain and retrieve program development information
f. Perform system generation -- create load tapes
g. Provide software management aids

4.1.3.1 FASP Usage Overview

This section is a general introduction to the use of FASP for software development on the NAVAIRDEVCEN CCS. Software development and maintenance is accomplished by submitting a computer job to the CCS. The job directs one or more FASP processing procedures to operate on the data base. The operations performed by FASP on the data base as a result of a series of computer jobs defines software development under FASP.

FASP runs under the CDC KRONOS operating system on the NAVAIRDEVCEN CCS. All 'work' accomplished on the CCS is a direct result of submitting a computer job which consists of a series of cards or statements that provide information to FASP which directs the operating system to perform certain functions. A FASP job may be logically divided into the following sections:

a. Job identification section. Identifies the job and provides accounting information required by the CCS.

b. FASP invocation section. Requests FASP be called and identifies the FASP procedures to be executed.

c. Data. Input data required by processing procedures.

d. End of job identifier.

As stated above, the FASP data base contains the developing software and information about the software. The maintenance and protection of the data base is one of the major responsibilities of the FASP system. FASP must insure that the data base is consistent, i.e., that the source library corresponds to the object library, that the data base is complete, i.e., all files are present, and that if the data base is neither consistent nor complete that the user can recover a 'good' data base. FASP protects the data base by keeping on-line and magnetic tape backup versions.

4.1.4 Conduct of QA Reviews and Audits

The most important task for QA is to assure that the QA policies, procedures, and software standards developed and identified in the appropriate documents are carried out. To this end, informal and formal QA reviews and audits are conducted incrementally on software products.

In the context in which they are employed at CSC, reviews are QA critiques performed against documents, whereas audits are QA critiques performed against processes. Audits are broader in scope than reviews and frequently include the review of specific documents. The criteria against which documents are reviewed are:

1. Adherence to format standards
2. Clarity of objectives
3. Technical content
4. Interdocument consistency
5. Traceability to higher level specifications.

All deliverable and other formal project documentation should be reviewed and approved by QA. This includes software development plans, work tasking and authorization procedures, configuration management plans, requirements specifications, design documents, design implementation plans, software standards, test plans and procedures, end item acceptance plans, and user's manuals.

QA audits usually include document reviews and in general perform four functions:

1. They assess compliance of source code and documentation with software standards and procedures;
2. They assure traceability of requirements;
3. They determine the satisfaction of system requirements during system test and acceptance phase;
4. They assess test sufficiency.

QA Audits are scheduled before a milestone event. The following is a sample of those that may be performed on software projects.

Software Requirements Audit. This audit is conducted prior to the software requirements review and includes the review of the requirements specifications, software development plan, and end item acceptance plan. QA reviews these documents to determine whether:

1. Software requirements are traceable to system requirements.
2. The planned development and test methods are adequate to assure satisfaction of software requirements.

Interface Verification Audit. The interface requirements and design specifications are audited as early as possible to identify and correct potential interface problems. The process of auditing an interface specification is similar to auditing a software requirements specification. QA personnel also participate in a weekly meeting of the interface control working group.

Preliminary and Detailed Design Audits. These audits should be conducted prior to PDR and CDR respectively, and are concerned with the format and content of the design documentation and test plans. Results of the findings are discussed during the PDR and CDR, respectively.

Incremental Development Audit. The incremental development audit approach involves the conduct of periodic audits during the code and unit test phase. When the "Build" concept is employed, these audits should coincide with the completion of each build. This allows the developers time to implement corrective actions in their Software Engineering Notebooks (SENs) and code without effecting cost and schedule performance. The use of checklists facilitates the audit as it standardizes the audit process and lets the developers know ahead of time what to expect. Incremental development audits include:

1. Verifying that changes to requirements, design, and interface definitions are maintained and updated.
2. Verifying that any applicable design or code walkthroughs were properly held.
3. Audit of source code for standards compliance.
4. Assessment of test sufficiency for thoroughness of unit testing.
5. Adequacy of configuration identification and control.

Pre-Turnover Audit. Prior to the turnover (release) of the software to the test team for formal testing, a pre-turnover audit should be conducted to assess the adequacy of unit testing. QA evaluates the total software product being turned over in terms of:

1. Actual vs. expected unit test output.
2. Change control status verification of code and documentation.
3. Completeness, content, and organization of SENs.
4. Code compliance to applicable software standards.
5. Data base control.
6. Test sufficiency.
7. Discrepancy report status.

This audit must be well planned and efficiently executed to minimize schedule impacts. Adequate time should be allotted to negotiate corrective action items and correct deficiencies in the software documentation prior to delivery to the test team. This audit is complete when QA certifies the condition of the software, and the test manager accepts the turnover package.

Test Audit. Software test audits should be conducted at the completion of each test phase. Their purpose is to:

1. Assure that the approved software configuration management procedures are being followed;
2. Assure that test procedures used by the test group are the current, approved versions;
3. Assure that test reports identify proper test procedures and software configuration, specify the test analysis and, if any deficiencies or deviations were noted, how they were explained and accounted for;
4. Verify that test procedures provide a step-by-step rationale for conducting a test, and that test results comply with acceptance criteria specified in the test procedure;
5. Verify that test data packages comply with approved formats. A test data package is a stand-alone set of documents for each test which contains test procedures, test reports, discrepancy reports, and test results. It facilitates an independent review of each test by QA and customer personnel.
6. Verify compliance by the test team with stated management procedures for change control, discrepancy reporting, test reporting, etc.
7. Verify the proper closure of each problem report.

This audit should not be confused with the inspection and test surveillance function described in the next section. The inspection and test surveillance function is a part of the daily QA involvement with the test activities and differs from an audit, which has contractual requirements and provides definite corrective actions and direction to the test activities.

At the conclusion of the final test phase, and perhaps as part of a physical configuration audit, software QA personnel verify the completion and presence of all software products. This certification is done prior to the turnover of the software to the TIES.

Management Procedures Audit. In addition to audits of technical material, periodic audits should be performed on the project management functions. Specific areas of interest include work tasking, security, cost/schedule control, deliverables, and adherence to the software development plan.

4.1.5 Inspection and Surveillance of Formal Tests

Inspection and test surveillance is an on-site activity performed by QA personnel during program testing. Specific tasks include:

1. Monitor all tests to ensure that actual tests performed are as specified in documented test procedures. This is accomplished via QA signoff on the test execution report.
2. Assure that all potential discrepancies are recorded in the approved manner.
3. Compare the configurations of hardware/software components utilized in the test against the configuration identified in the test procedure.
4. Certify that analysis of the test output is correct, the test satisfies its intended requirements and acceptance criteria, and that the test data package is complete.
5. Assure that master copies of test procedures, test results, and test reports are maintained and available from a centralized records center.

4.1.6 Configuration Verification

New versions of programs are usually released after formal configuration control procedures have been applied. An important QA function is to perform configuration audits on the controlled master libraries of each version to assure that no inadvertent or unauthorized code changes have been made. One method being used to carry out

this function is the following three-step procedure: (1) isolate the approved modifications, (2) delete those modifications from the new master file and replace any removed code (this can be done automatically by a library management system), and (3) compare the old file and the rearranged new file (an operating system utility does this most efficiently). They should be identical. Any discrepancy should be brought to the immediate attention of the configuration control manager for corrective action. A letter certifying the authenticity of the new file may accompany the new release notice. Other configuration verifications may be incoming and outgoing shipment packages where QA personnel inspect the product packages to assure that the contents are as specified.

4.1.7 Management of the Problem Reporting System

An important management control mechanism is the accurate recording, tracking, and reporting of all real and perceived software problems. The QA organization with its project wide visibility is often the prime candidate for this job. As problem trouble reports (PTRs) are written, a copy should be logged. As is often the case, the original PTR gets sent to configuration control which decides whether it should be rejected, deferred, or accepted, and if accepted who should fix it. QA can track each PTR and report its status at any point in time. Periodic error trend analyses can then be made which report, for example:

1. The error rate for various programs.
2. The number of open PTRs.
3. The relative frequency of PTRs in various defect categories.
4. The average time to close various problems.

4.1.8 Retention of QA Records

The previous seven QA functions often create voluminous paper, much of it contractual in nature. To properly retain and control this paper, a repository with formal procedures should be established early in the project life cycle. On small projects this may involve only a secretary with a locked file cabinet, while on larger projects QA may make use of a centralized data management center. In the quality planning phase during proposal or contract definition, the appropriate QA records should be defined and resources budgeted for this control. These records may include:

1. QA Plans
2. QA Procedures
3. Design Problem Reports
4. Software Problem Reports
5. QA Audits and Reviews
6. Test Data Packages
7. Waivers and Deviations.

APPENDIX A

TIES DOCUMENTATION

TITLE	AUTHOR/DESCRIPTION
APPLICATION OF A SAMPLE MOTOROLA MC68488 INTEGRATED CIRCUIT TO IMPLEMENTATION OF AN IEEE 688-1975 INTERFACE	Chester M. Nowicki Naval Air Development Center Warminster, Pennsylvania 18974
TACTICAL INFORMATION EXCHANGE SYSTEM Interim Engineering Report No. 3 (Period 3 August 1977 - October 1977)	Ford Aerospace & Communications Corporation Western Development Laboratories Division
TECHNICAL MEMORANDUM TIES "BIG BOARD"	Naval Air Development Center Warminster, Pennsylvania 18974 4042 10 April 1979
TIES SOFTWARE ORIENTATION AND REQUIREMENTS	Naval Air Development Center Warminster, Pennsylvania 18974
TIES STANDARD DIGITAL CONTROL INTERFACES	Chester M. Nowicki Naval Air Development Center Warminster, Pennsylvania 18974
TIES TACTICAL INFORMATION EXCHANGE SYSTEM	G.J. Palatucci and E.R. Ressler Naval Air Development Center Warminster, Pennsylvania 18974
SYSTEMATIC SOFTWARE DEVELOPMENT FOR CONTROL OF INSTRUMENTS ON THE IEEE-488 BUS	Leon Smith Naval Air Development Center Warminster, Pennsylvania

LANGUAGE DOCUMENTATION

TITLE	AUTHOR/DESCRIPTION
FORTH'S FORTE IS TIGHTER PROGRAMMING	Electronics, March 15, 1979 Copyright 1979 by
HOL EVALUATION FOR MISSILE SOFTWARE APPLICATIONS	U.S. Army Missile Command Guidance and Control Directorate Missile Computer Software & Hardware Center Redstone Arsenal, Alabama 35809
RATIONALE FOR THE DESIGN OF THE GREEN PROGRAMMING LANGUAGE	
REFERENCE MANUAL FOR THE GREEN PROGRAMMING LANGUAGE	
PASCAL USER MANUAL AND REPORT	Kathleen Jensen Niklaus Wirth
PASCAL ISN'T JUST ONE MORE COMPUTER LANGUAGE. IT PROMISES TO BE SIMPLE, FLEXIBLE AND FAST.	Electronic Design 19, September 13, 1978
POWERFUL STATEMENTS AND A VERSATILE SYNTAX GIVE A PASCAL PROGRAM "BODY"	Electronic Design 20, September 27, 1978
PASCAL'S INPUT AND OUTPUT PROCEDURES ARE POWERFUL, YET EASY TO MASTER	Electronic Design 23, November 8, 1978
VERSATILE GROUPINGS AND USER-DEFINED DATA TYPES MAKE PASCAL REALLY SHINE	Electronic Design 23, November 8, 1978
MOVING BIT PATTERNS OR WORD GROUPS CAN BE A STRUGGLE - BUT NOT IN PASCAL	Electronic Design 25, December 6, 1978
WITH A REAL-TIME OPERATING SYSTEM, A PASCAL PROGRAM CAN RUN YOUR TEST SET	Electronic Design 26, December 20, 1978

MISCELLANEOUS DOCUMENTS

TITLE	AUTHOR/DESCRIPTION
MILITARY STANDARD TACTICAL SOFTWARE DEVELOPMENT	Proposed MIL-STD-1679 (Navy)
THE DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS	Alfred V. Ahe Bell Laboratories John E. Hopcroft Cornell University
DIGITAL SYSTEMS DEVELOPMENT METHODOLOGY	Copyright 1978 by Computer Sciences Corporation
DOCUMENTATION FROM THE SOFTWARE MANAGEMENT CONFERENCE 1978	
TACTICAL DIGITAL SYSTEMS DOCUMENTATION STANDARDS	SECNAVINST 3560.1 8 August 1974

APPENDIX C

THE INITIALIZATION CODE FOR THE SYSTEM CONTROLLER

THE INITIALIZATION CODE SEGMENT OF THE SYSTEM CONTROLLER

The System Controller controls TIES processing, handles TIES configuration, reconfiguration, man machine interface, function simulation, show and tell, bit, and error recovery.

The SysC data base consists of the Resource Table, Band-Mode Table, and the Mode Encoding Table. The Mode Encoding Table is discussed in detail in the Preliminary Performance Specification (Table 3.3.5.1) and is stored in character string W$ in the program. The resource table and band-mode table are described in Appendix A and tables 3.3.5.2 and 3.3.5.3.

Input can be optionally READ in as DATA, or INPUT from the TEXTRON keyboard, and the program can be easily modified to do this (with one minor restriction). Details on input are given in Appendix B.

The meaning of error messages printed out during the running at the program are given in Appendix C. A detailed discussion of the system controller data base is given in section 3.3.5.1.1 of the PPS.

The code is listed in Appendix D.

APPENDIX A

RESOURCE TABLE DESCRIPTION:

ARRAY R1(I, J) Contains data as follows:

J	Description
1.	Primary Resource Address
2.	Section Address
3.	Second Address
4.	ASCII of Status
5.	ASCII of "F" or "S"
6.	Remaining Transmit Capacity
7.	Remaining Receive Capacity
8.	Position in mode encoding table of first band contained in primary and section address resource.
9.	Product of prime numbers encoding modes corresponding to band the position of which is in location J = 9 of array .

Notes

7 + 2 *(B1). Position is made encoding table of last band contained in primary and section address resource.

8 + 2 *(B1). Product of prime numbers encoding modes corresponding to band the position of which is in location J = 7 + 2 *(B1) of array.

17. ASCII of "T" or "R" obtained from band-mode table matching there the band, primary and section addresses, and mode in location 18 of the Ith resource.

BAND-MODE TABLE DESCRIPTION

ARRAY: M1 (I, J) contains data as follows.

J	Description
1.	Position in mode encoding table corresponding to band name.
2.	Position in mode encoding table corresponding to mode name.
3.	Transmit FDM channel.
4.	Receive FDM channel.
5.	ASC ("F")
6.	ASC ("R")
7.	Primary address of resource for F and R.
8.	Section address of resource for F and R.
9.	ASC ("F")
10.	ASC ("T")
11.	Primary address of resource for F and T.
12.	Section address of resource for F and T.
13.	ASC ("S")
14.	ASC ("R")
15.	Primary address of resource for S and R.
16.	Section address of resource for S and R.
17.	ASC ("S")
18.	ASC ("T")
19.	Primary address of resource for S and T.
20.	Section address of resource for S and T.

ARRAY Q$:

Holds parameter strings - each one up to 40 characters in length.

APPENDIX B

INPUT FORMAT

All of the following READ instructions can be changed (optionally) to INPUT from the keyboard.

READ:	Variable	Instruction Sequence
Max. number of resources	R	2010
Max. number of bands	M	2010
Currently needed No. of resources	R2	2250
Resource class data for Ith resource	$(1 \leq I \leq R2)$	
Primary Address	R1(I, 1)	2280
Section Address	R1(I, 2)	2280
Second Address	R1(I, 3)	2280
"F" or "S" (Frequency or Signal)	N$	2310
Remaining Transmit Capacity	R1(I, 6)	2330
Remaining Receive Capacity	R1(I, 7)	2330
Number of Bands for Resource I	B1	2340
Band Names (See Note 1)	C$	4110
Encoded (Product of Primes) Mode Names	R1(I, 8 + 2*B2)	2400
Band-mode data for Ith Band $(1 \leq I < M)$ (See Note 2)		
Band Name (See Notes 1 and 2)	C$	4110
Mode Name	E$	2490
For each Band-Mode		
K = 0: Primary Address for Freq.-Receive M1 (I, 7)		3770

K = 0:	Section Address for Freq. Receive M1(I, 8)	3770
K = 0:	Parameters for Freq. - Receive (Note 3) P$	3780
K = 1:	Primary Address for Freq. - Transmit M1(I, 11)	3770
K = 1:	Section Address for Freq. - Transmit M1(I, 12)	3770
K = 1:	Parameters for Freq. - Transmit (Note 3) P$	3780
K = 2:	Primary Address for Signal - Receive M1(I, 15)	3770
K = 2:	Section Address for Signal - Receive M1(I, 16)	3770
K = 2:	Parameters for Signal - Receive (Note 3) P$	3780
K = 3:	Primary Address for Signal - Transmit M1(I, 19)	3770
K = 3:	Section Address for Signal - Transmit M1(I, 20)	3770
K= 3:	Parameters for Signal - Transmit (Note 3) P$	3780

Note 1: Since the same address is used, Band names must be READ for both resource and band-name tables or INPUT for both resource and band-name tables. READ/INPUT mixture is not permitted for band names.

Note 2: If Band name = "END", READ-INPUT of band mode data is terminated.

Note 3: Parameters have a maximum of 40 characters for each combination of F/S and R/T for each I.

18. Prime number corresponding to first mode obtained from band-mode table matching there the band and primary and section addresses for the Ith resource.

15 + 2 *Z1. ASCII of "T" or "R" obtained from band-mode table matching there the band, primary and section addresses, and mode in location 16 + 2 * Z1 of the Ith resource.

16 + 2 *Z1. Prime number corresponding to last mode obtained from band-mode table matching there the band and primary and section addresses for the Ith resource.

Maximum value of B1 = 4; of Z1=25.

APPENDIX C

ERROR MESSAGES:

"TOO MANY MODES": The maximum value allowed for M is 38.

"MORE RESOURCES NEEDED THAN FIT IN ARRAY": Variable R2, the number of resources needed > R, max number of resources.

"ILLEGAL MODE": Mode does not exist for band read in - see encoding table.

"RESOURCE IN MODE INPUT NOT FOUND IN RESOURCE INPUT": There is no resource address pair (Primary and Section addresses) which was read or input into the resource table that matches, for both the primary and section address, this pair that was just read or input into the band-mode table.

"BAND NOT FOUND FOR THIS RESOURCE": For each primary and section address pair resource read or input into the resource table, none, one, or some bands may have been read or input into this table and so associated with it. The band name read or input into the band-mode table has resource address pairs read or in at associated with it. For one of these pairs identical to the resource address pair in the resource table, no band name is found associated with it that is the same as the band name in the band-mode table.

"MODE DOES NOT EXIST IN BAND": The prime number that corresponds to the mode read into the band-mode table does not divide evenly into the product of primes, corresponding to the set of modes, for the same band and primary and section addresses in the resource table.

"FOR SAME PRIMARY AND SECTION ADDRESSES": Primary and Section address indicators.

"RESOURCE AND MODE INPUTS F/S DO NOT AGREE": The F/S for the primary and section addresses read into the resource table is "F" or "S". The F/S for the same pair of primary and section addresses read into the mode table is the other of "F" or "S".

APPENDIX D

CODE LISTING AND FLOWCHARTS

```
100 REM
110 REM THIS IS THE A PRIORI DATA
120 REM -------------------------
130 REM MAX PROGRAM CAPACITY FOR
150 REM RESOURCES
160 REM MODES
170 REM PARAMETERS
190 DATA 20,38
200 REM
210 REM NUMBER OF RESOURCE CLASSES
220 REM
230 DATA 17
240 REM
250 REM RESOURCE CLASS DATA
260 REM
270 REM--------NOTES-------
280 REM FORMAT OF RESOURCE CLASS DATA IS AS FOLLOWS
290 REM DATA PRIMARY uP ADDRESS, SECTION ADDRESS, SECOND uP ADDRESS
300 REM DATA TYPE, REMAINING TRANSMIT CAPACITY, REMAINING RECEIVE
310 REM      CAPACITY, NUMBER OF FREQUENCIES
320 REM DATA LISTS OF MODES HANDLED, WITH MODE ENCODING
330 REM
340 DATA 1,1,5
350 DATA "F",0,1,1
360 DATA "LX",2310
370 DATA 1,2,5
380 DATA "F",0,1,1
390 DATA "LX",2310
400 DATA 1,3,5
410 DATA "F",0,1,1
420 DATA "LX",2310
430 DATA 1,4,5
440 DATA "F",0,1,1
450 DATA "LX",2310
```

```
460 DATA 1,5,5
470 DATA "F",96,0,1
480 DATA "LX",2310
490 DATA 2,0,0
500 DATA "F",98,1,2
510 DATA "UHF",2310
520 DATA "VHF",30
530 DATA 3,0,0
540 DATA "F",98,1,2
550 DATA "UHF",2310
560 DATA "VHF",30
570 DATA 4,0,0
580 DATA "F",99,1,1
590 DATA "HF",210
600 DATA 5,1,1
610 DATA "S",1,1,1
620 DATA "LX",7
630 DATA 5,2,1
640 DATA "S",1,1,1
650 DATA "LX",330
660 DATA 5,3,1
670 DATA "S",1,1,1
680 DATA "LX",330
690 DATA 5,4,1
700 DATA "S",1,1,1
710 DATA "LX",330
720 DATA 6,1,0
730 DATA "S",1,1,3
740 DATA "UHF",210
750 DATA "VHF",30
760 DATA "HF",42
770 DATA 6,2,8
780 DATA "S",1,1,3
790 DATA "UHF",210
800 DATA "VHF",30
```

```
810 DATA "HF",42
820 DATA 6,3,0
830 DATA "S",1,1,3
840 DATA "UHF",210
850 DATA "VHF",30
860 DATA "HF",42
870 DATA 7,1,0
880 DATA "S",1,1,3
890 DATA "UHF",2310
900 DATA "VHF",30
910 DATA "HF",210
920 DATA 7,2,0
930 DATA "S",1,1,3
940 DATA "UHF",2310
950 DATA "VHF",30
960 DATA "HF",210
970 REM MODE PRESETS
980 REM FOR GIVEN BELOW
990 REM DATA BAND, MODE ID
1000 REM DATA TRANSMIT FDM CHANNEL, RECEIVE FDM CHANNEL
1010 REM DATA FCU RECEIVE
1020 REM       PRIMARY uP ADDRESS
1030 REM       SECTION CODE
1040 REM       PARAMETERS
1050 REM DATA FCU TRANSMIT
1060 REM       PRIMARY uP ADDRESS
1070 REM       SECTION CODE
1080 REM       PARAMETERS
1090 REM DATA SCU RECEIVE
1100 REM       PRIMARY uP ADDRESS
1110 REM       SECTION ADDRESS
1120 REM       PARAMETER
1130 REM DATA SCU TRANSMIT
1140 REM       PRIMARY u@ ADDRESS
1150 REM       SECTION ADDRESS
```

```
1160 REM        PARAMETERS
1170 DATA "LX","JTIDS"
1180 DATA 1,2
1190 DATA 1,1,"PARAMETERS TO SETUP FCU RECEIVE"
1200 DATA 1,5,"PARAMETERS TO SET UP FCU TRANSMIT"
1210 DATA 5,1,"PARAMETERS TO SET UP SCU RECEIVE"
1220 DATA 5,1,"PARAMETERS TO SET UP SCU TRANSMIT"
1230 DATA "LX","IFF"
1240 DATA 3,4
1250 DATA 1,3,"PARAMETERS TO SET UP FCU RECEIVE"
1260 DATA 1,5,"PARAMETERS TO SET UP FCU TRANSMIT"
1270 DATA 5,2,"PARAMETERS TO SET UP SCU RECEIVE"
1280 DATA 5,2,"PARAMETERS TO SET UP SCU TRANSMIT"
1290 DATA "LX","TACAN"
1300 DATA 5,6
1310 DATA 1,4,"PARAMETERS TO SET UP FCU RECEIVE"
1320 DATA 1,5,"PARAMETERS TO SET UP FCU TRANSMIT"
1330 DATA 5,3,"PARAMETERS TO SET UP SCU RECEIVE"
1340 DATA 5,3,"PARAMETERS TO SET UP SCU TRANSMIT"
1350 DATA "UHF","AM VOICE"
1360 DATA 7,8
1370 DATA 2,0,"PARAMETERS TO SET UP FCU RECEIVE"
1380 DATA 2,0,"PARAMETERS TO SET UP FCU TRANSMIT"
1390 DATA 6,1,"PARAMETERS TO SET UP SCU RECEIVE"
1400 DATA 6,1,"PARAMETERS TO SET UP SCU TRANSMIT"
1405 DATA "VHF","FM DATA"
1410 DATA 9,10
1420 DATA 3,0,"PARAMETERS TO SET UP FCU RECEIVE"
1430 DATA 3,0,"PARAMETERS TO SET UP FCU TRANSMIT"
1440 DATA 6,2,"PARAMETERS TO SET UP SCU RECEIVE"
1450 DATA 6,2,"PARAMETERS TO SET UP SCU TRANSMIT"
1460 DATA "HF","LINK 11"
1480 DATA 11,12
1490 DATA 4,0,"PARAMETERS TO SET UP FCU RECEIVE"
1500 DATA 4,0,"PARAMETERS TO SET UP FCU TRANSMIT"
```

```
1510 DATA 7,1,"PARAMETERS TO SET UP SCU RECEIVE"
1520 DATA 7,2,"PARAMETERS TO SET UP SCU TRANSMIT"
1530 DATA "END"
1540 REM 1400,1405 #ING PROBLEM
1550 B$="   LX UHFVHFHF "
1810 L$="DABS     GPS       IFF       JTIDS    TACAN    "
1820 U$="AM VOICEFM DATA FM VOICEFSK      LINK 4   "
1830 V$="AM VOICEFM DATA FM VOICE                  "
1840 H$="AM       AM <SSB>LINK 11 TTY/FSK          "
1850 DIM W$(160)
1860 W$=L$&U$
1870 W$=W$&V$
1880 W$=W$&H$
1890 PRINT "IS RESOURCE DISPLAY REPORT DESIRED? YES OR NO"
1900 INPUT X$
1910 IF X$="NO" THEN 2010
1920 IF X$<>"YES" THEN 1890
1930 PAGE
1940 PRINT "         RESOURCE DISPLAY REPORT"
1950 PRINT "         -------- ------- ------"
1960 PRINT " "
1970 PRINT "             F  R  R T"
1980 PRINT "           S /  T  R /"
1990 PRINT "ADDRESS    T S  C  C R  BAND MODE(S)"
2000 PRINT "-------    - -  -  - -  ---- -------"
2010 READ R,M
2020 IF M>38 THEN 3400
2030 DELETE R1
2040 DELETE M1
2050 DIM R1(R,26),M1(M,20),P9(5)
2060 P9(1)=2
2070 P9(2)=3
2080 P9(3)=5
2090 P9(4)=7
2100 P9(5)=11
```

```
2110 M1=0
2120 R1=0
2130 REM INITIALIZE MATRICES FOR RESOURCES,MODES
2140 DELETE Q$
2150 DIM Q$(6144)
2160 LET Q$="                              "
2170 Q$=Q$&Q$
2180 Q$=Q$&Q$
2190 Q$=Q$&Q$
2200 Q$=Q$&Q$
2210 Q$=Q$&Q$
2220 Q$=Q$&Q$
2230 Q$=Q$&Q$
2240 Q$=Q$&Q$
2250 READ R2
2260 IF R2>R THEN 3730
2270 FOR I=1 TO R2
2280 READ R1(I,1),R1(I,2),R1(I,3)
2290 REM PRIMARY,SECTION,SECOND ADDRESSES
2300 REM
2310 READ N$
2320 R1(I,5)=ASC(N$)
2330 READ R1(I,6),R1(I,7)
2340 READ B1
2350 REM  B1 NUMBER OF BANDS
2360 FOR B2=1 TO B1
2370 GOSUB 4110
2380 REM BAND NAME
2390 R1(I,7+2*B2)=D
2400 READ R1(I,8+2*B2)
2410 REM ENCODED MODE NAMES
2420 NEXT B2
2430 NEXT I
2440 REM NOW FOR MODES
2450 FOR I=1 TO M
```

```
2460 GOSUB 4110
2470 IF C$="END" THEN 2770
2480 M1(I,1)=D
2490 READ E$
2500 Z=LEN(E$)
2510 FOR K=0 TO 4
2520 A$=SEG(W$,40*(D-1)+K*8+1,Z)
2530 IF E$=A$ THEN 2570
2540 NEXT K
2550 PRINT "ILLEGAL MODE"
2560 GO TO 4230
2570 M1(I,2)=K+1
2580 E=P9(K+1)
2590 M2=I
2600 READ M1(I,3),M1(I,4)
2610 REM TRANSMIT AND RECEIVE FDM CHANNELS
2620 K=0
2630 M1(I,5)=ASC("F")
2640 M1(I,6)=ASC("R")
2650 GOSUB 3750
2660 M1(I,9)=ASC("F")
2670 M1(I,10)=ASC("T")
2680 GOSUB 3750
2690 M1(I,13)=ASC("S")
2700 M1(I,14)=ASC("R")
2710 GOSUB 3750
2720 M1(I,17)=ASC("S")
2730 M1(I,18)=ASC("T")
2740 GOSUB 3750
2750 NEXT I
2770 IF X$="NO" THEN 3420
2775 C=8
2780 FOR I=1 TO R2
2790 E$="?"
2800 N$=CHR(R1(I,5))
```

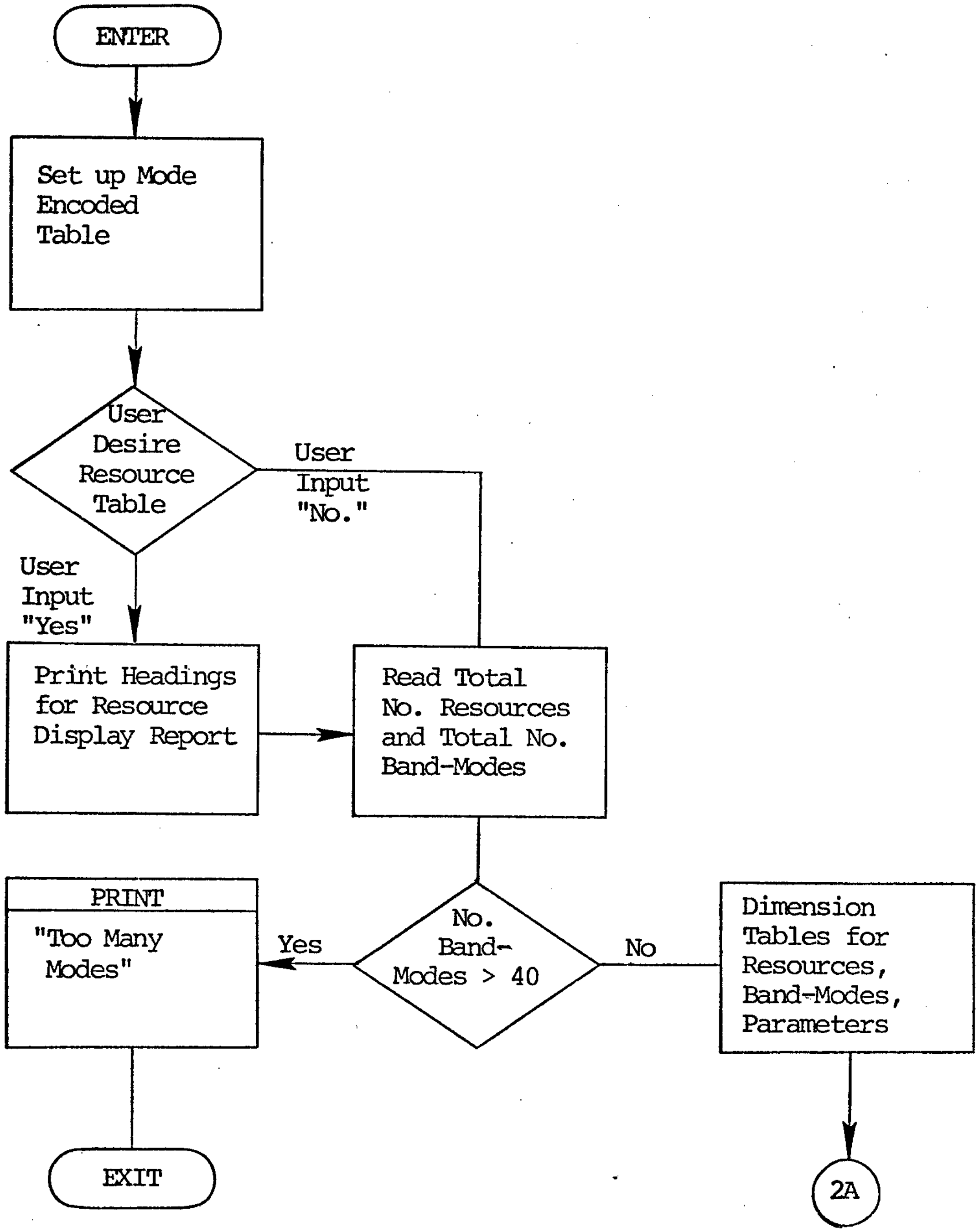
```
2810 FOR Z=17 TO 25 STEP 2
2820 IF R1(I,Z)=ASC("T") THEN 2930
2830 NEXT Z
2840 REM FOLLOWING NO TRANSMITS
2850 FOR Z=17 TO 25 STEP 2
2860 IF R1(I,Z)=ASC("R") THEN 3220
2870 NEXT Z
2880 REM FOLLOWING NO T OR R
2890 PRINT USING 2910:R1(I,1),R1(I,2),R1(I,3),E$,N$,R1(I,6),R1(I,7)
2895 C=C+1
2900 GO TO 3350
2910 IMAGE 2D,X;D,X,2D,2X,A,X,A,X,2D,X,2D
2920 REM NOW TO PRINT OUT TRANSMITS
2930 FOR B2=1 TO B1
2940 IF R1(I,7+2*B2)<>0 THEN 2960
2950 NEXT B2
2960 L$="  "
2970 K$=SEG(B$,3*R1(I,7+2*B2)+1,3)
2980 L$=K$&L$
2990 K$=L$
3000 FOR Z1=Z TO 25 STEP 2
3010 IF R1(I,Z1)<>ASC("T") THEN 3050
3020 M$=SEG(W$,40*(R1(I,7+2*B2)-1)+8*(R1(I,Z1+1)-1)+1,8)
3030 M$=M$&" "
3040 L$=L$&M$
3050 NEXT Z1
3060 PRI USI 3070:R1(I,1),R1(I,2),R1(I,3),E$,N$,R1(I,6),R1(I,7),"T",L$
3065 C=C+1
3070 IMAGE 2D,X,D,X,2D,2X,A,X,A, X,2D,X,2D,X,A,2X,50A
3080 FOR Z=17 TO 25 STEP 2
3090 IF R1(I,Z)=ASC("R") THEN 3130
3100 NEXT Z
3110 GO TO 3350
3120 REM LAST INSTRUC IF NO RECEIVE
3130 FOR Z1=Z TO 25 STEP 2
```

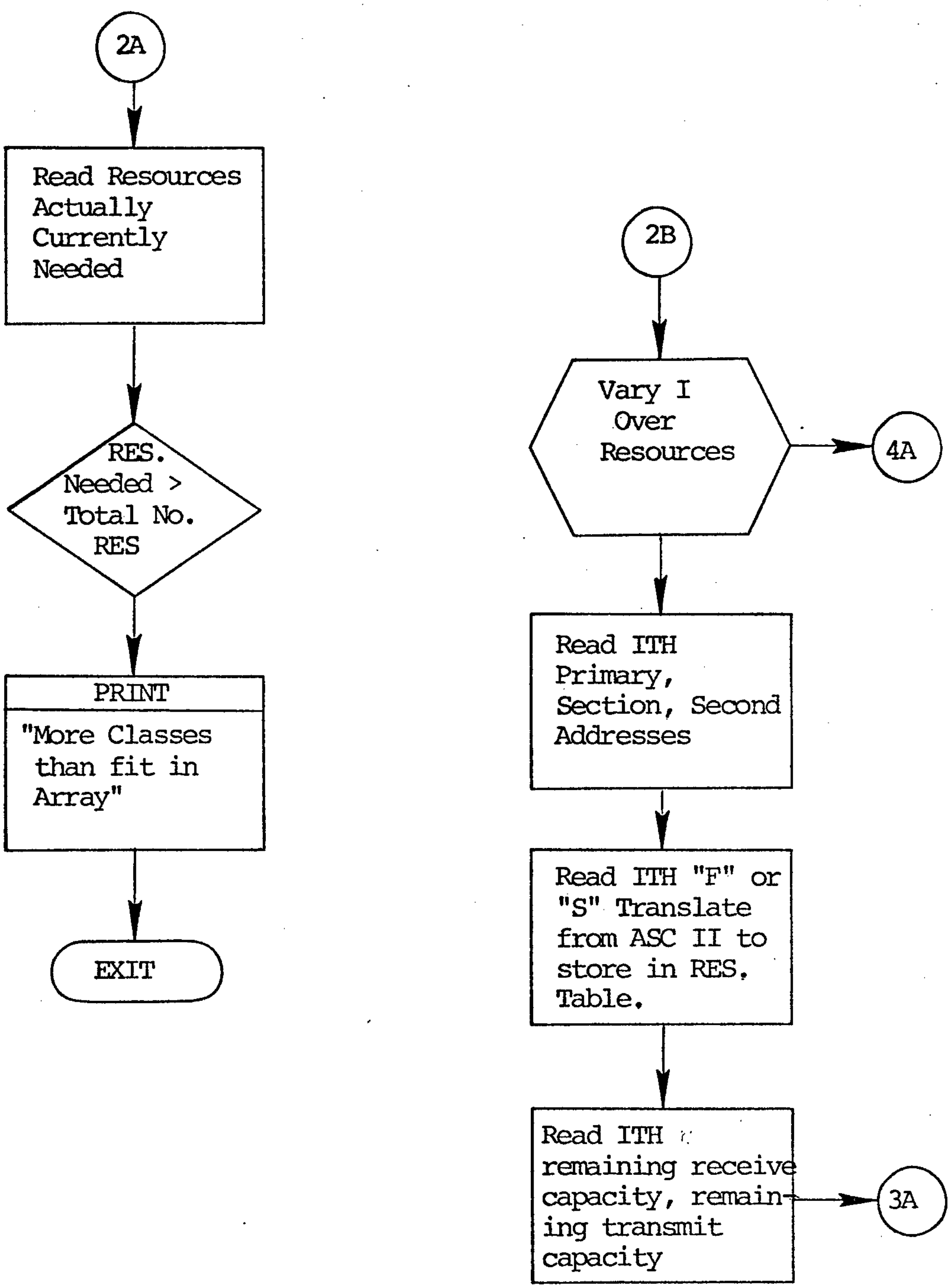
```
3140 IF R1(I,Z1)<>ASC("R") THEN 3180
3150 M$=SEG(W$,40*(R1(I,7+2*B2)-1)+8*(R1(I,Z1+1)-1)+1,8)
3160 M$=M$&" "
3170 K$=K$&M$
3180 NEXT Z1
3190 PRINT USING 3200:"R",K$
3195 C=C+1
3200 IMAGE 19X,A,2X,51A
3210 GO TO 3350
3220 FOR B2=1 TO B1
3230 IF R1(I,7+2*B2)<>0 THEN 3250
3240 NEXT B2
3250 L$="  "
3260 K$=SEG(B$,3*R1(I,7+2*B2)+1,3)
3270 L$=K$&L$
3280 FOR Z1=Z TO 25 STEP 2
3290 IF R1(I,Z1)<>ASC("R") THEN 3330
3300 M$=SEG(W$,40*(R1(I,7+2*B2)-1)+8*(R1(I,Z1+1)-1)+1,8)
3310 M$=M$&" "
3320 L$=L$&M$
3330 NEXT Z1
3340 PRI USI 3070:R1(I,1),R1(I,2),R1(I,3),E$,N$,R1(I,6),R1(I,7),"R",L$
3345 C=C+1
3350 IF I=R2 THEN 3380
3360 IF R1(I+1,1)=R1(I,1) THEN 3380
3370 PRINT " "
3375 C=C+1
3380 IF C<29 THEN 3388
3382 PRINT "MORE"
3384 C=1
3388 NEXT I
3390 GO TO 3420
3400 PRINT "TOO MANY MODES"
3410 GO TO 4230
3420 REM NOW FOR MODES
```

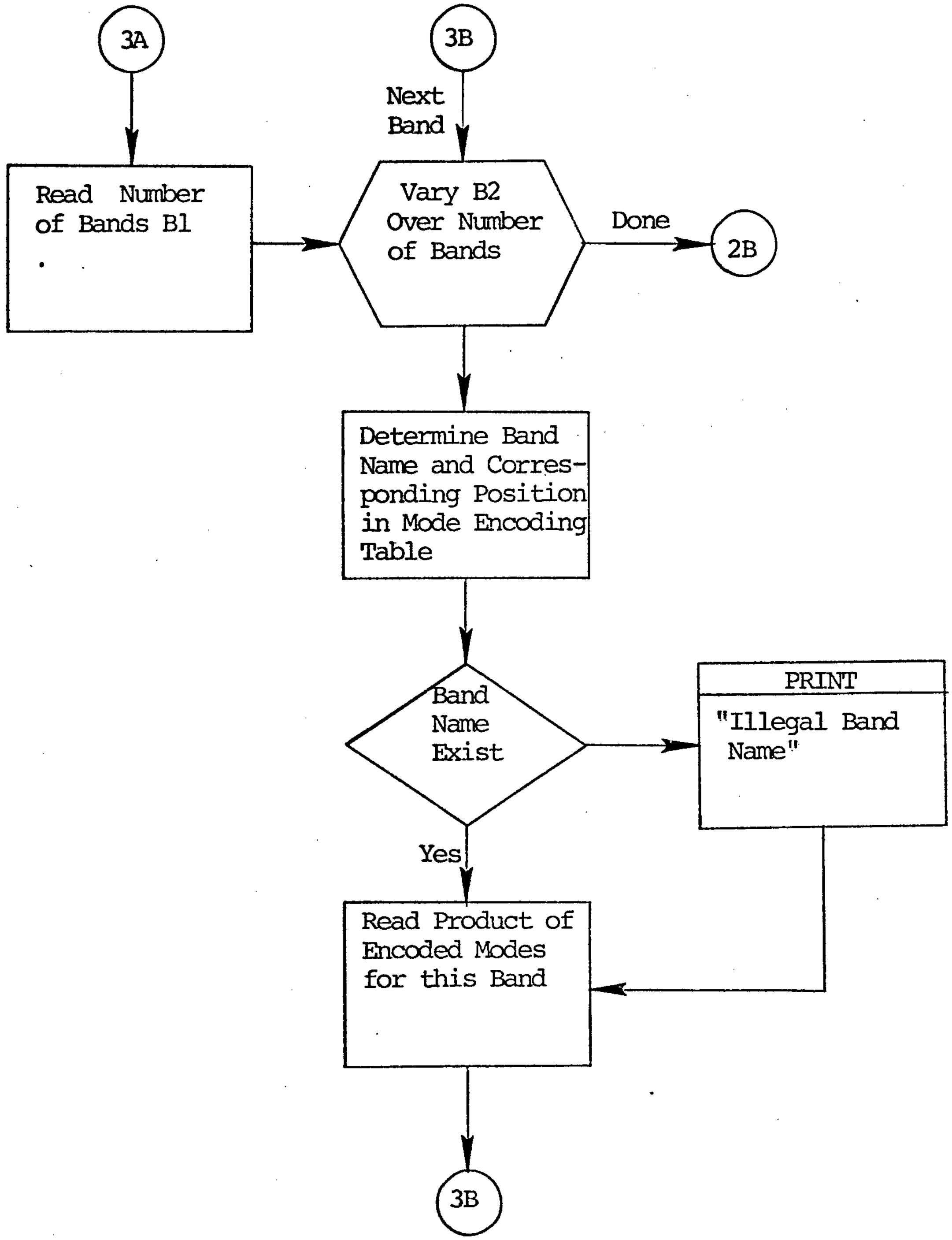
```
3430 PRINT "IS MODE DISPLAY REPORT DESIRED? YES OR NO"
3440 INPUT X$
3450 IF X$="NO" THEN 4230
3460 IF X$<>"YES" THEN 3430
3470 PAGE
3480 PRINT "        MODE DISPLAY REPORT"
3490 PRINT "        ---- ------- ------"
3500 PRINT "                         S"
3510 PRINT "                 R F T   E"
3520 PRINT "              T  E / /   C"
3530 PRINT "BAND MODE     R  C S R P T PARAMETERS"
3540 PRINT "---- ----     -  - - - - - ----------"
3550 FOR I=1 TO M2
3560 D=M1(I,1)
3570 A$=SEG(B$,3*D+1,3)
3580 M$=SEG(W$,40*(D-1)+8*(M1(I,2)-1)+1,8)
3590 REM M$ MODE NAME
3600 P$=SEG(Q$,160*(I-1)+1,40)
3610 PRINT USING 3620:A$,M$,M1(I,3),M1(I,4),"F","R",M1(I,7),M1(I,8),P$
3620 IMAGE 3A,X,8A,3D,3D,X,A,X,A,2D,2D,X,40A
3630 P$=SEG(Q$,160*(I-1)+41,40)
3640 PRINT USING 3690:"F","T",M1(I,11),M1(I,12),P$
3650 P$=SEG(Q$,160*(I-1)+81,40)
3660 PRINT USING 3690:"S","R",M1(I,15),M1(I,16),P$
3670 P$=SEG(Q$,160*(I-1)+121,40)
3680 PRINT USING 3690:"S","T",M1(I,19),M1(I,20),P$
3690 IMAGE 19X,A,X,A,2D,2D,X,40A
3700 PRINT " "
3710 NEXT I
3720 GO TO 4230
3730 PRINT " MORE RESOURCES NEEDED THAN FIT IN ARRAY"
3740 GO TO 4230
3750 REM
3760 REM PRIMARY AND SECTION ADDRESSES AND PARAMETERS
3770 READ M1(I,7+4*K),M1(I,8+4*K)
```

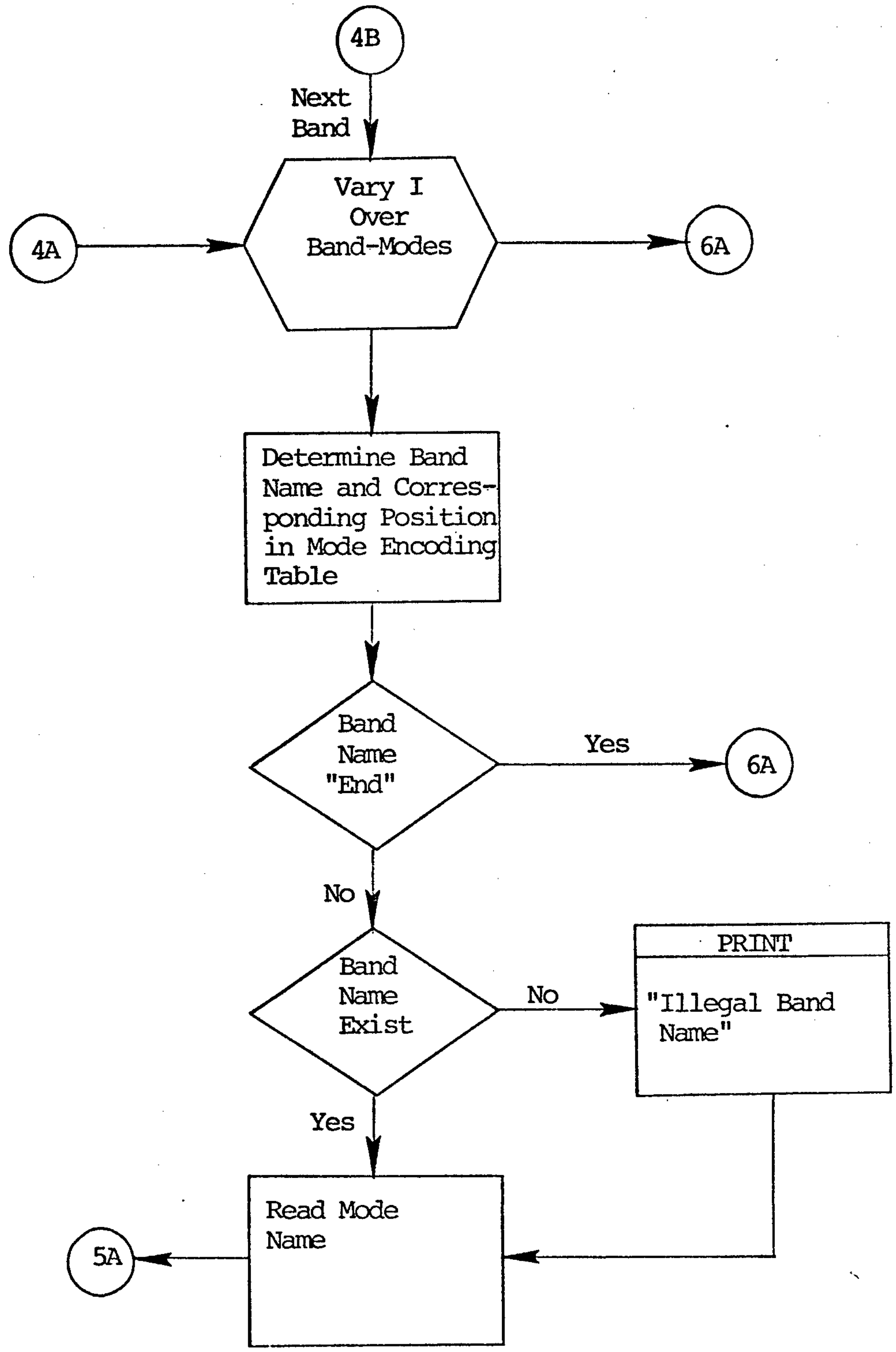
```
3780 READ P$
3790 P$=SEG(P$,1,40)
3800 Q$=REP(P$,160*(I-1)+40*K+1,40)
3810 FOR J=1 TO R2
3820 IF M1(I,7+4*K)<>R1(J,1) THEN 3840
3830 IF M1(I,8+4*K)=R1(J,2) THEN 3870
3840 NEXT J
3850 PRINT "RESOURCE IN MODE INPUT NOT FOUND IN RESOURCE INPUT"
3860 GO TO 4090
3870 FOR B2=1 TO B1
3880 IF D=R1(J,7+2*B2) THEN 3920
3890 NEXT B2
3900 PRINT "BAND NOT FOUND FOR THIS RESOURCE"
3910 GO TO 4090
3920 FOR B3=1 TO B1
3930 IF B3=B2 THEN 3950
3940 R1(J,7+2*B3)=0
3950 NEXT B3
3960 IF R1(J,8+2*B2)=INT(R1(J,8+2*B2)/E)*E THEN 3990
3970 PRINT "MODE DOES NOT EXIST IN BAND"
3980 GO TO 4090
3990 FOR Z=18 TO 26 STEP 2
4000 IF R1(J,Z)=0 THEN 4030
4010 NEXT Z
4020 REM STORE ASSOCIATED MODE IN RESOURCE TABLE
4030 R1(J,Z)=M1(I,2)
4040 IF R1(J,5)=M1(I,5+4*K) THEN 4080
4050 PRINT "FOR SAME PRIMARY AND SECTION ADDRESSES"
4060 PRINT R1(J,1),R1(J,2)
4070 PRINT "RESOURCE AND MODE INPUTS F/S DO NOT AGREE"
4080 R1(J,Z-1)=M1(I,6+4*K)
4090 K=K+1
4100 RETURN
4110 REM DETERMINE BAND NAME AND POSITION
4120 READ C$
```

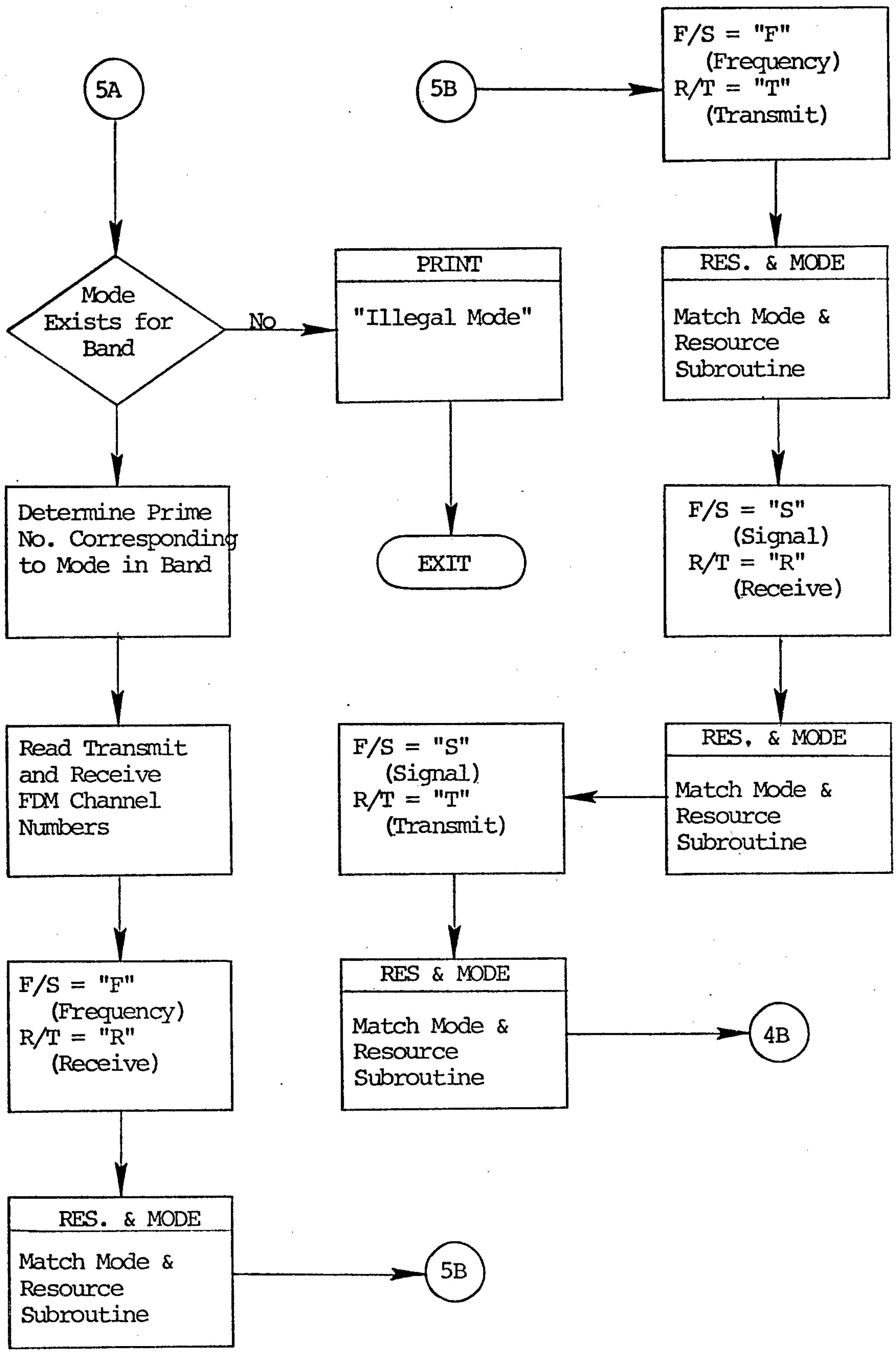
```
4130 IF C$="END" THEN 4220
4140 C=LEN(C$)
4150 IF C=3 THEN 4170
4160 C$=C$&" "
4170 FOR D=1 TO 4
4180 A$=SEG(B$,3*D+1,3)
4190 IF C$=A$ THEN 4220
4200 NEXT D
4210 PRINT "ILLEGAL BAND NAME"
4220 RETURN
4230 END
```

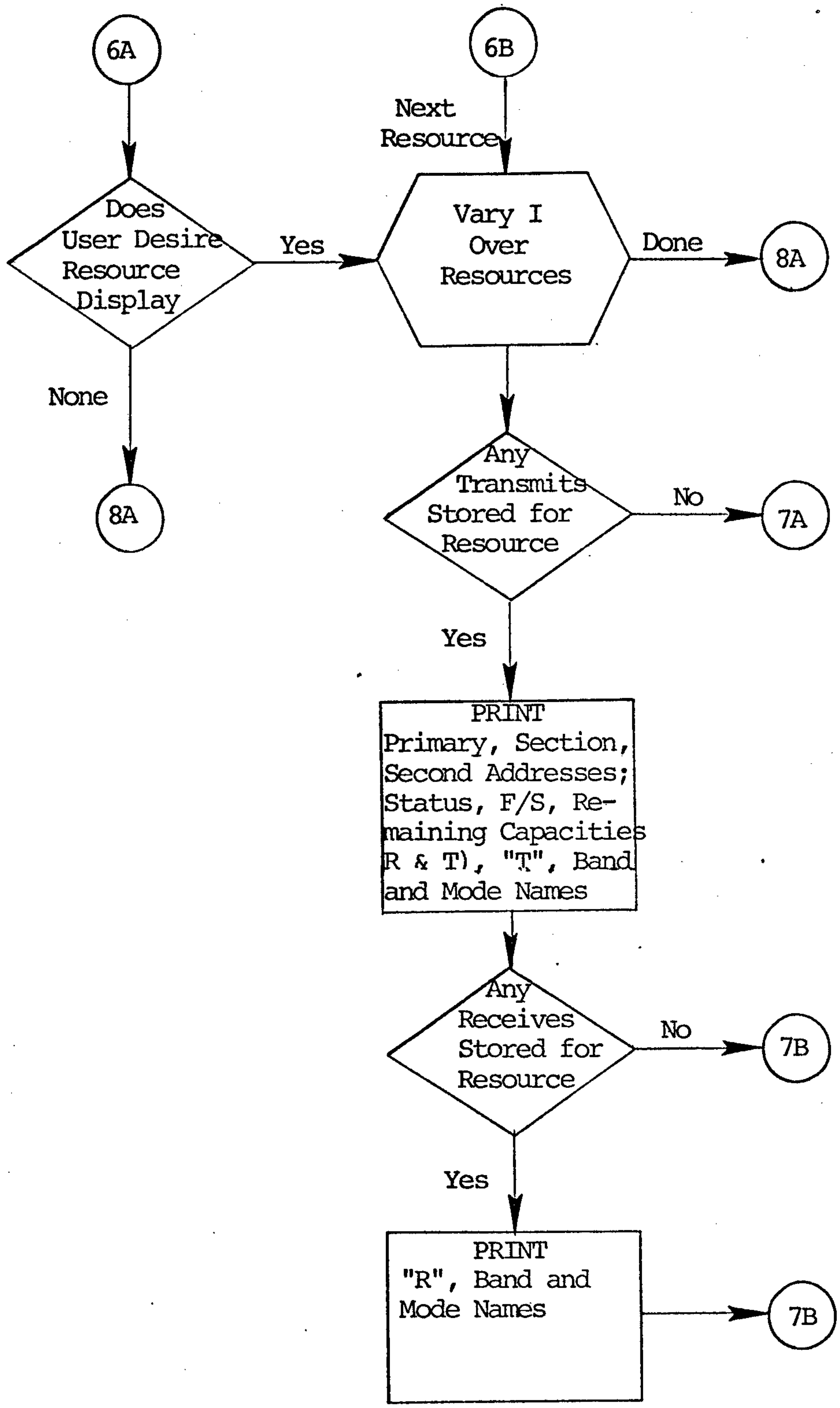

ENTER
Set up Mode Encoded Table
User Desire Resource Table
User Input "No."
User Input "Yes"
Print Headings for Resource Display Report
Read Total No. Resources and Total No. Band-Modes
PRINT
"Too Many Modes"
Yes
No. Band-Modes > 40
No
Dimension Tables for Resources, Band-Modes, Parameters
EXIT
2A

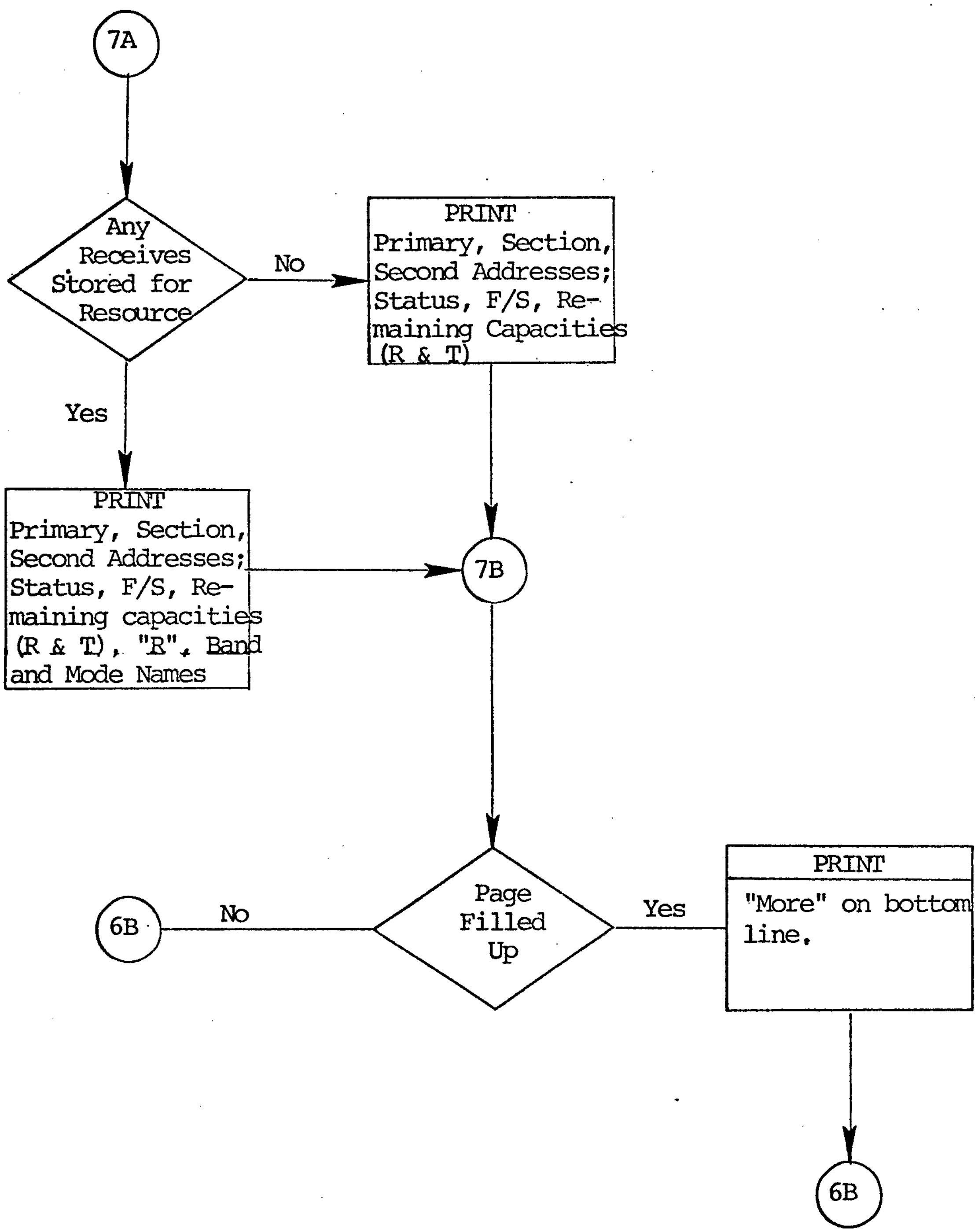

2A
Read Resources Actually Currently Needed
RES. Needed > Total No. RES
PRINT
"More Classes than fit in Array"
EXIT
2B
Vary I Over Resources
4A
Read ITH Primary, Section, Second Addresses
Read ITH "F" or "S" Translate from ASC II to store in RES. Table.
Read ITH remaining receive capacity, remaining transmit capacity
3A

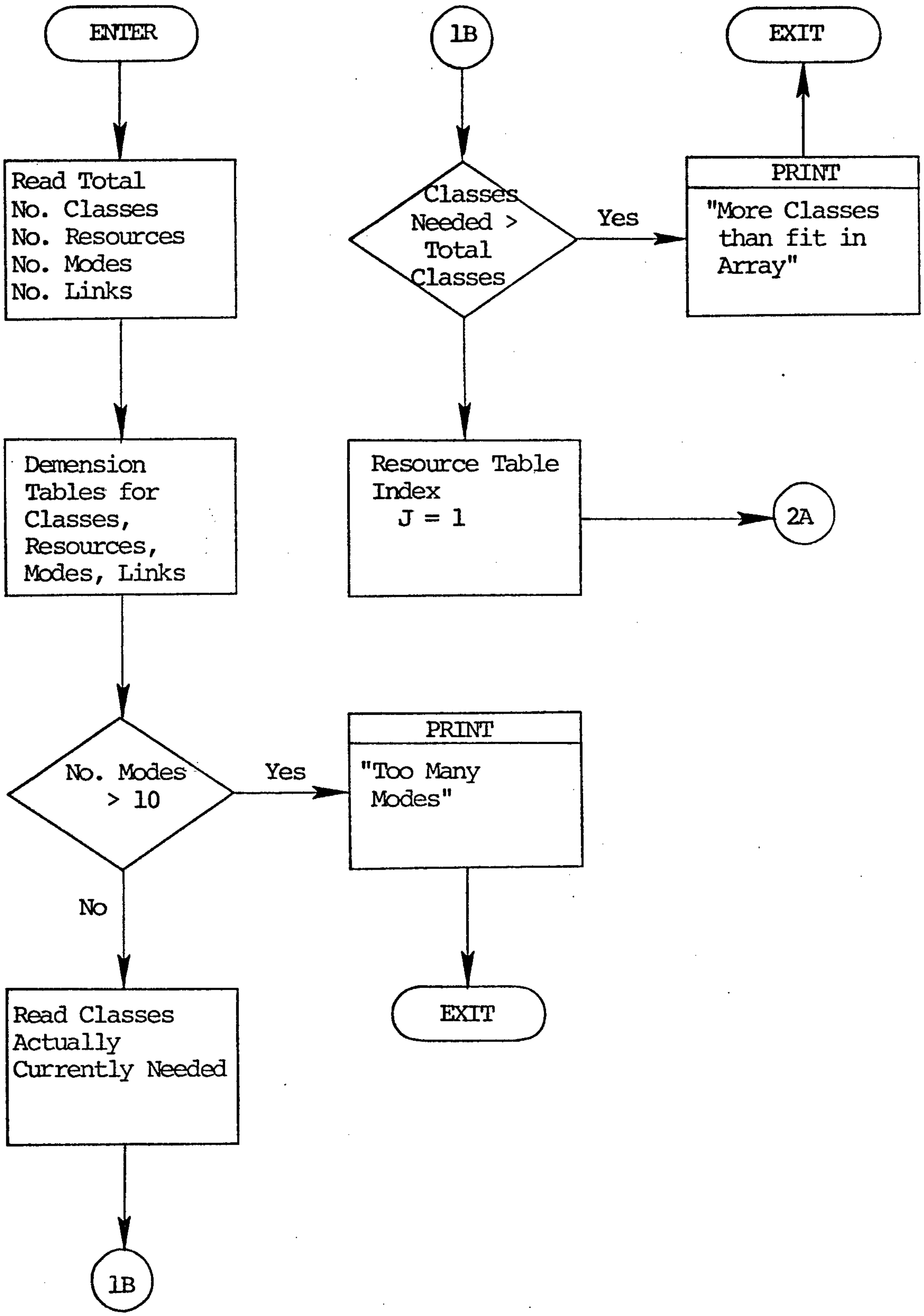

3A
3B
Next Band
Read Number of Bands B1
Vary B2 Over Number of Bands
Done
2B
Determine Band Name and Corresponding Position in Mode Encoding Table
Band Name Exist
PRINT
"Illegal Band Name"
Yes
Read Product of Encoded Modes for this Band
3B

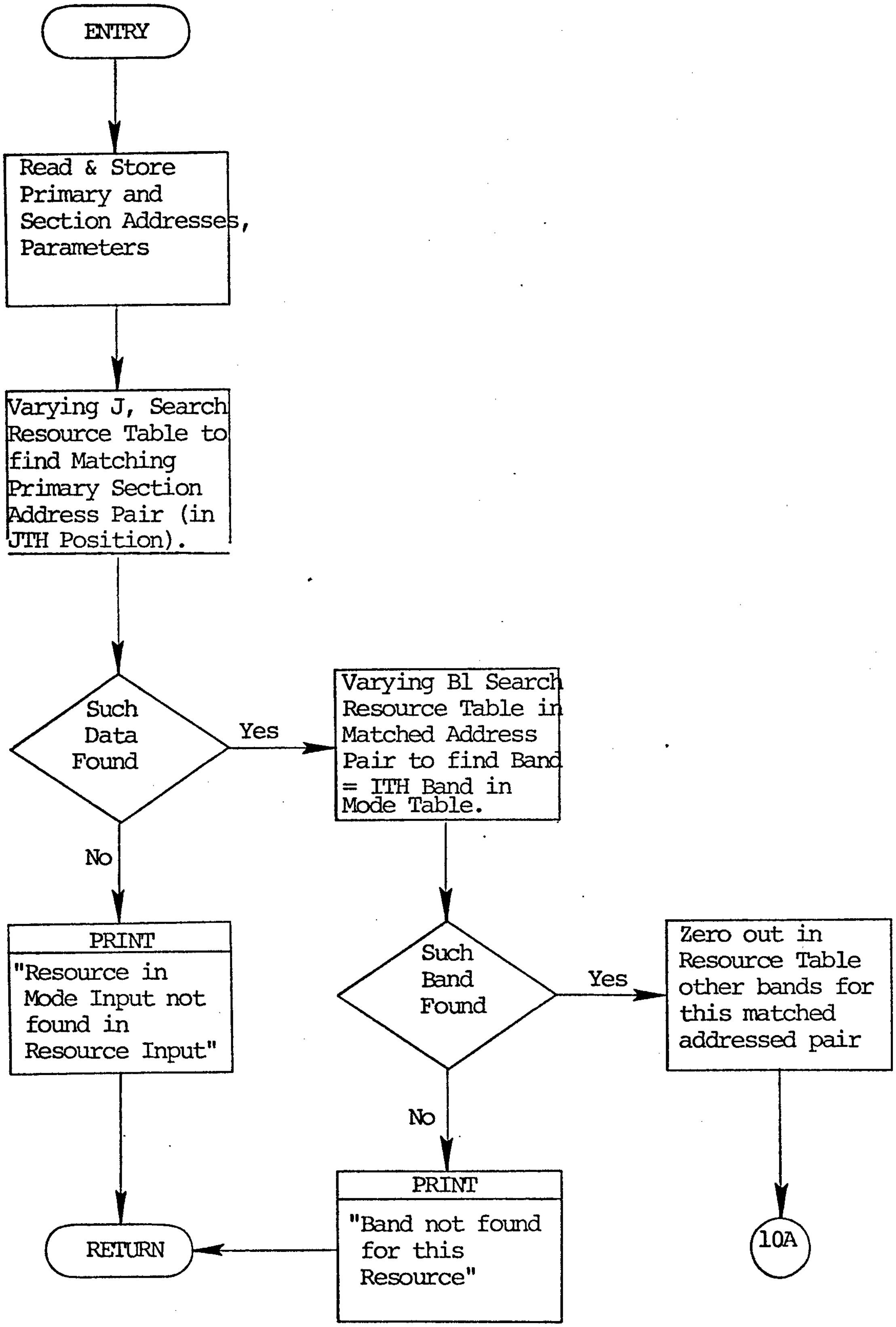

4B
Next Band
Vary I Over Band-Modes
4A
6A
Determine Band Name and Corresponding Position in Mode Encoding Table
Band Name "End"
Yes
6A
No
Band Name Exist
No
PRINT
"Illegal Band Name"
Yes
Read Mode Name
5A

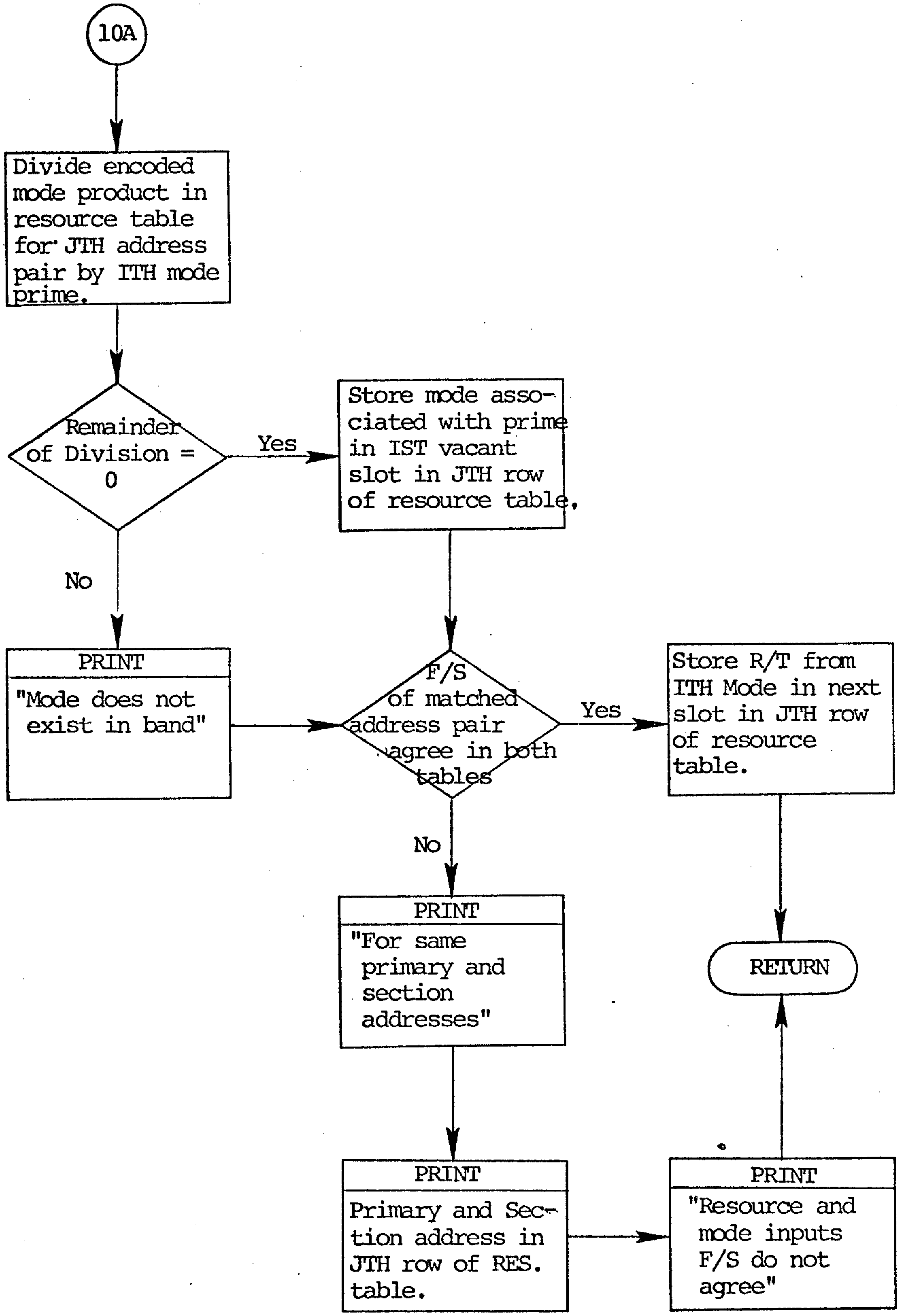

5A
Mode Exists for Band
No
PRINT
"Illegal Mode"
EXIT
Determine Prime No. Corresponding to Mode in Band
Read Transmit and Receive FDM Channel Numbers
F/S = "F" (Frequency) R/T = "R" (Receive)
RES. & MODE
Match Mode & Resource Subroutine
5B
5B
F/S = "F" (Frequency) R/T = "T" (Transmit)
RES. & MODE
Match Mode & Resource Subroutine
F/S = "S" (Signal) R/T = "R" (Receive)
RES. & MODE
Match Mode & Resource Subroutine
F/S = "S" (Signal) R/T = "T" (Transmit)
RES & MODE
Match Mode & Resource Subroutine
4B


6A
Does User Desire Resource Display
Yes
None
8A
6B
Next Resource
Vary I Over Resources
Done
8A
Any Transmits Stored for Resource
No
7A
Yes
PRINT Primary, Section, Second Addresses; Status, F/S, Remaining Capacities R & T), "T", Band and Mode Names
Any Receives Stored for Resource
No
7B
Yes
PRINT "R", Band and Mode Names
7B


7A
Any Receives Stored for Resource
No
PRINT
Primary, Section, Second Addresses; Status, F/S, Remaining Capacities (R & T)
Yes
PRINT
Primary, Section, Second Addresses; Status, F/S, Remaining capacities (R & T), "R", Band and Mode Names
7B
6B
No
Page Filled Up
Yes
PRINT
"More" on bottom line.
6B


ENTER
1B
EXIT
Read Total
No. Classes
No. Resources
No. Modes
No. Links
Classes
Needed >
Total
Classes
Yes
PRINT
"More Classes
than fit in
Array"
Demension
Tables for
Classes,
Resources,
Modes, Links
Resource Table
Index
J = 1
2A
No. Modes
> 10
Yes
PRINT
"Too Many
Modes"
No
Read Classes
Actually
Currently Needed
EXIT
1B


ENTRY
Read & Store Primary and Section Addresses, Parameters
Varying J, Search Resource Table to find Matching Primary Section Address Pair (in JTH Position).
Such Data Found
Yes
No
Varying Bl Search Resource Table in Matched Address Pair to find Band = ITH Band in Mode Table.
PRINT
"Resource in Mode Input not found in Resource Input"
Such Band Found
Yes
No
Zero out in Resource Table other bands for this matched addressed pair
PRINT
"Band not found for this Resource"
RETURN
10A


10A
Divide encoded mode product in resource table for JTH address pair by ITH mode prime.
Remainder of Division = 0
Yes
Store mode associated with prime in IST vacant slot in JTH row of resource table.
No
PRINT
"Mode does not exist in band"
F/S of matched address pair agree in both tables
Yes
Store R/T from ITH Mode in next slot in JTH row of resource table.
No
PRINT
"For same primary and section addresses"
RETURN
PRINT
Primary and Section address in JTH row of RES. table.
PRINT
"Resource and mode inputs F/S do not agree"